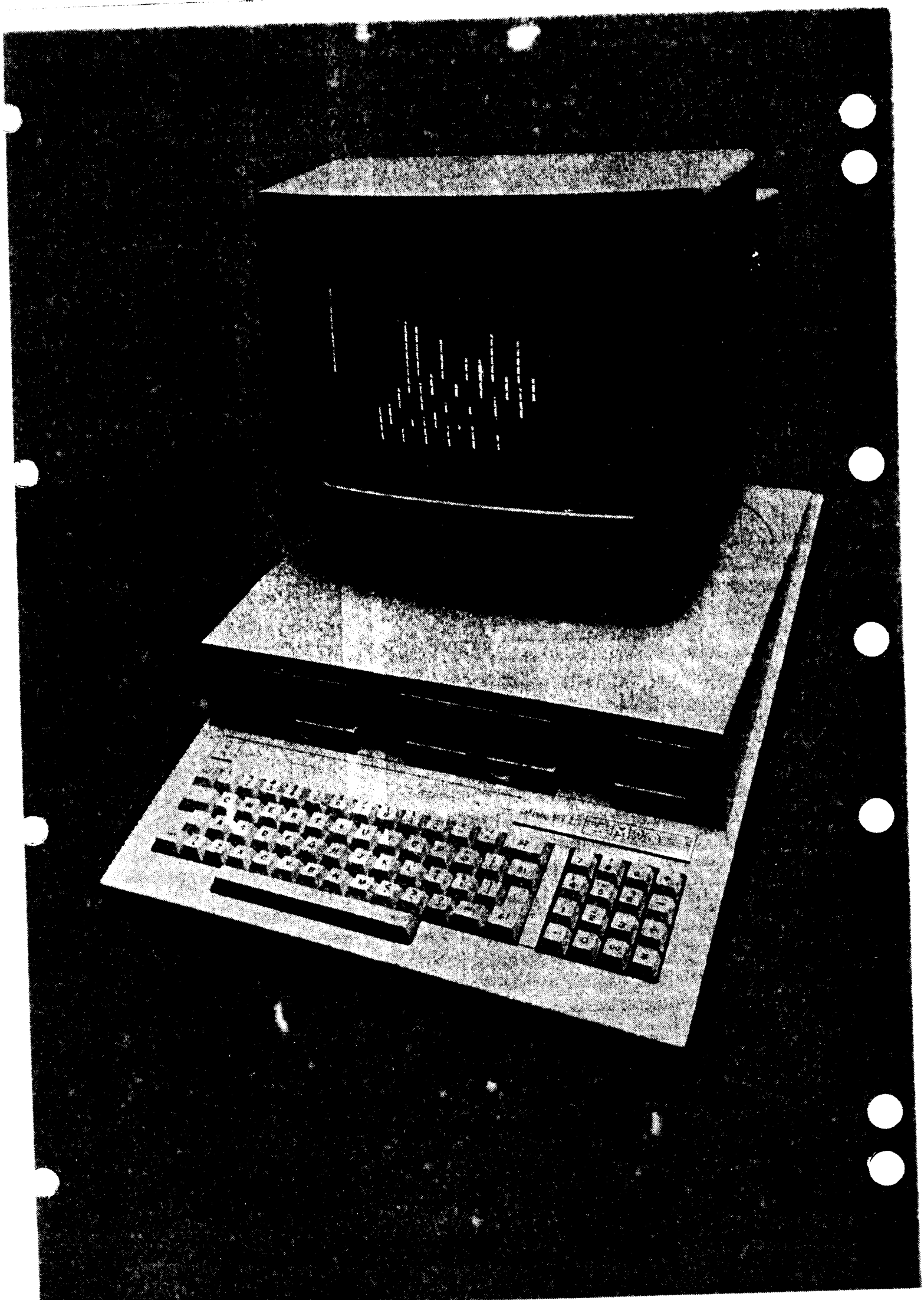
M20

ISAM
(Index Sequential Access Method)

Guida Utente

olivetti L1



LIBRERIA PERIFERICHE DI SISTEMA

- MANUALE PER L'INSTALLAZIONE — 3966260 V
- INTRODUZIONE AL SISTEMA — 3930100 Y (1)
- STAMPANTE PR 2400 GUIDA OPERATIVA — 3984140 T (1)
- STAMPANTE PR 1450 GUIDA OPERATIVA — 3984160 V (1)
- STAMPANTE PR 1471 GUIDA OPERATIVA — 3984180 F (1)
- STAMPANTE PR 1481 GUIDA OPERATIVA (*) — 3984230 U
- STAMPANTE PR 430 GUIDA OPERATIVA (*) — 3982930 A

LIBRERIA SOFTWARE APPLICATIVO

- MULTIPLAN GUIDA DELL'UTENTE — 3983380 N
- OLIWORD GUIDA DELL'UTENTE (*) — 3983400 U
- OLIENTRY GUIDA DELL'UTENTE (*) — 3983420 J
- OLIMASTER USER GUIDE — 3983430 K
- OLISTAT (Statistical Analysis) USER GUIDE — 3983450 M
- OLINC USER GUIDE (*) — 3983470 P
- OLISORT GUIDA DELL'UTENTE — 3982250 D
- OLINUM (Numerical Analysis) USER GUIDE — 3987570 J

LIBRERIA PROGRAMMI

- OLITUTOR USER GUIDE — 3987610 N
- PCOS (Professional Computer Operating System) GUIDA DELL'UTENTE — 3987580 T
- LINGUAGGIO BASIC MANUALE GENERALE — 3982420 N (1)
- BASIC & PCOS COMPENDIO ISTRUZIONI — 3987630 Q
- LINGUAGGIO ASSEMBLER MANUALE GENERALE — 3982400 Y
- LINGUAGGIO ASSEMBLER GUIDA DELL'UTENTE — 3987680 K
- LINGUAGGIO ASSEMBLER COMPENDIO ISTRUZIONI (*) — 3983540 N
- FORTRAN MANUALE GENERALE (*) — 3987720 R
- PASCAL MANUALE GENERALE (*) — 3987700 T

LIBRERIA FUNZIONI AVANZATE

- OLITEST GUIDA DELL'UTENTE — 3982320 C
- I/O CON PERIFERICHE ESTERNE GUIDA DELL'UTENTE — 3982290 R
- VIDEOTEX GUIDA DELL'UTENTE (*) — 3982270 F
- OLICOM GUIDA DELL'UTENTE — 3983670 K
- OLITERM GUIDA DELL'UTENTE — 3983700 S

(*) I manuali tras. sono planificati per Release successive

PREFAZIONE

Questa guida utente descrive le funzionalità di ISAM (Index Sequential Access Method) ed il loro uso da parte di programmi BASIC su M20.

Il capitolo 1 descrive i concetti generali del metodo di accesso con chiave ai file dati ed é indirizzato soprattutto ai lettori che hanno poca esperienza di tecniche di accesso ISAM. Esso fornisce un'introduzione alla parte successiva del manuale dove sono illustrate le strutture dati usate da ISAM ed é descritto l'uso delle sue funzioni elementari da parte di un programma BASIC.

L'appendice A descrive un programma didattico per l'utilizzo del package ISAM e contiene esempi che hanno lo scopo di rendere più familiari al lettore le tecniche usate.

La presenza di esempi di programmi BASIC richiede la conoscenza del linguaggio BASIC da parte del lettore, oltre ad una sua familiarità con l'M20.

RIFERIMENTI:

Linguaggio BASIC
Manuale Generale
Codice 3982420 N (1)

Professional Computer
Operating System (PCOS)
Guida dell'Utente
Codice 3987580 T (0)

DISTRIBUZIONE: Generale (G)

PRIMA EDIZIONE: Ottobre 1982

RELEASE: 1.3 e successive

OLICOM, OLITERM, OLIWORD, OLINUM, OLISTAT, OLITEST, GTL, OLITUTOR, OLIENTRY, OLISORT, OLIMASTER, sono marchi della Ing. C. Olivetti & C. S.p.A.

MULTIPLAN è un marchio registrato della MICROSOFT Inc.

© 1982, by Olivetti

PUBBLICAZIONE EMESSA DA:

Ing. C. Olivetti & C., S.p.A.
Servizio Centrale Documentazione
77, Via Jervis - 10015 IVREA

INDICE

1. CONCETTI GENERALI PER LA GESTIONE DEI FILE CON CHIAVE

METODI DI GESTIONE DEI FILE 1-1

STRUTTURE DI FILE CON CHIAVE 1-1

INDICI SECONDARI 1-2

CHIAVI CONCATENATE 1-3

LETTURA DI RECORD CON CHIAVE 1-3

CANCELLAZIONE DI RECORD E DI CHIAVE 1-5

ESEMPI DI USO DI FILE CON CHIAVE 1-5

INTERROGAZIONE DI ARCHIVI 1-5

CONTROLLI 1-6

AGGIORNAMENTI 1-6

INSERIMENTI E CANCELLAZIONI 1-7

2. STRUTTURA DI FILE ISAM

FILE ISAM 2-1

IL FILE DATI 2-1

IL FILE INDICE 2-1

STRUTTURA B TREE 2-2

LA STRUTTURA ISAM AD ALBERO 2-4

BLOCCHI DI CONTROLLO DATI 2-5

NUMERO DI FILE APERTI E FILE BUFFERING 2-6

RECORD CANCELLATI 2-7

3. USO DI ISAM IN UN PROGRAMMA BASIC

FILE PROGRAMMA ISAM 3-1

USO DI ISAM SU M20 3-1

USO DI ISAM IN UN PROGRAMMA BASIC 3-1

COMUNICAZIONE CON ISAM 3-2

CODICI DI RITORNO 3-8

FUNZIONI DI UTILITA' ISAM 3-10

FUNZIONE DI STATO (MS) 3-10

FUNZIONE DI INIZIALIZZAZIONE (MI) 3-11

APERTURA E CHIUSURA DI FILE ISAM 3-13

APERTURA DI FILE (OO) 3-13

CREAZIONE DI FILE (OC) 3-14

APERTURA/CREAZIONE DI FILE (OF) 3-15

CHIUSURA DI FILE (CL) 3-16

LETTURA DATI DA UN FILE ISAM 3-18

LETTURA CON CHIAVE (RK, SK) 3-18

LETTURA GENERICA (RG, SG) 3-20

LETTURA SEQUENZIALE (RN, SN) 3-23

LETTURA SEQUENZIALE ALL'INDIETRO (RP, SP) 3-25

INSERIMENTO DI DATI IN UN FILE ISAM 3-27

INSERIMENTO ADD (WA, SA) 3-27

FUNZIONI DI CANCELLAZIONE 3-31

CANCELLAZIONE DI CHIAVE (KD, SD) 3-31

CANCELLAZIONE DI RECORD (DR) 3-33

A. PROGRAMMA DIDATTICO ISAM

B. PROGRAMMA DI CONTROLLO DEI FILE ISAM

C. VARIABILI ISAM

VARIABILI PASSATE AD ISAM C-1

VARIABILI RITORNATE DA ISAM C-2

D. CODICI DI FUNZIONE

E. CODICI DI RITORNO

1. CONCETTI GENERALI PER LA GESTIONE DEI FILE CON CHIAVE

SOMMARIO

Questo capitolo fornisce un'introduzione generale alla gestione di file con chiave. Esso descrive i tipi di file usati, i metodi con i quali l'informazione può essere trasferita dai file e fornisce alcuni esempi di situazioni nelle quali l'uso di file con chiave può rivelarsi particolarmente utile.

INDICE

METODI DI GESTIONE DEI FILE 1-1

STRUTTURE DI FILE CON CHIAVE 1-1

INDICI SECONDARI 1-2

CHIAVI CONCATENATE 1-3

LETTURA DI RECORD CON CHIAVE 1-3

CANCELLAZIONE DI RECORD E DI CHIAVE 1-5

ESEMPI DI USO DI FILE CON CHIAVE 1-5

INTERROGAZIONE DI ARCHIVI 1-5

CONTROLLI 1-6

AGGIORNAMENTI 1-6

INSERIMENTI E CANCELLAZIONI 1-7

CONCETTI GENERALI PER LA GESTIONE DEI FILE CON CHIAVE

METODI DI GESTIONE DEI FILE

Un sistema di gestione di file con chiave permette di accedere ai record dei file sia in modo sequenziale che in modo diretto (random) sulla base di una chiave (o dato identificatore) del singolo record dati.
ISAM (Index Sequential Access Method) è un esempio di tale sistema.

Prima di procedere ad un'analisi più dettagliata della gestione dei file con chiave consideriamo brevemente due fra i più noti e semplici metodi di accesso:

- metodo di accesso sequenziale (SAM): è un metodo in cui i record vengono letti e/o scritti uno dopo l'altro in sequenza fisica. Ogni qualvolta un record viene creato, esso è registrato alla fine del file. Per inserire un record all'interno di un file bisogna riscrivere tutto il file, collocando il nuovo record nella posizione desiderata. La lettura di un record può essere effettuata solamente mediante scansione sequenziale del file fino a che il record viene trovato. Aggiornamenti di record possono essere ammessi o meno, a seconda dell'implementazione del metodo di accesso.

- metodo di accesso diretto (DAM): è anche detto metodo "random". Esso permette la scrittura e/o lettura di record specificando la posizione fisica del record su disco oppure il numero del record (posizione assoluta o relativa) all'interno del file.

  Un record può quindi essere letto direttamente conoscendo o il numero del record stesso (posizione fisica sul disco) o la posizione relativa rispetto ad alcuni dati nel record.

  Nell'indirizzamento relativo il numero del record può essere usato come chiave del record stesso.

Entrambi questi metodi presentano vantaggi per particolari tipi di applicazioni. La struttura a file con chiave si rivela particolarmente utile ove è necessario accedere ai record sia in modo sequenziale che in modo random.

STRUTTURE DI FILE CON CHIAVE

Esistono due tipi principali di strutture dati associate ad un file con chiave: il file indice ed il file dati.

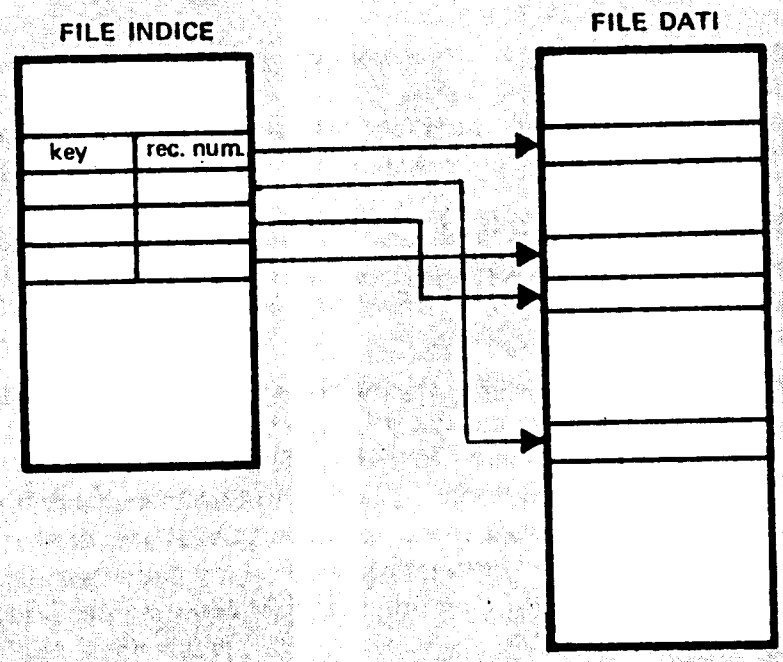


Figura 1-1 File indice e file dati

Il file indice è una sequenza di record contenenti la chiave ("Key") del record dati ed un puntatore ("pointer") al record stesso. La chiave identifica univocamente il record dati. Possono anche esistere indici secondari o alternativi che forniscono vie di accesso diversificate al file dati.

E' anche possibile avere la stessa chiave ripetura più volte all'interno di un file indice. Ad esempio, se il nome di una persona viene usato come chiave, possono esserci più persone con lo stesso nome. Allo scopo di mantenere accesso unico a tutti i record nel file occorre quindi, in casi come questo, specificare il numero relativo del record ("record number") all'interno dell'insieme dei record aventi la stessa chiave.

Il file dati contiene i record dati. Esso può essere organizzato in base al valore della chiave oppure in modo diverso a seconda della tecnica di implementazione.

INDICI SECONDARI

Gli indici secondari permettono l'accesso ai record per mezzo di chiavi diverse da quella principale (primaria). Ad esempio i record appartenenti ad un file clienti possono essere letti specificando o il numero di codice del cliente oppure il suo nome.

L'indice primario verrà quindi costruito usando il numero di codice come chiave, l'indice secondario userà il nome come chiave secondaria. In questo modo abbiamo due accessi al file diversi ed indipendenti.

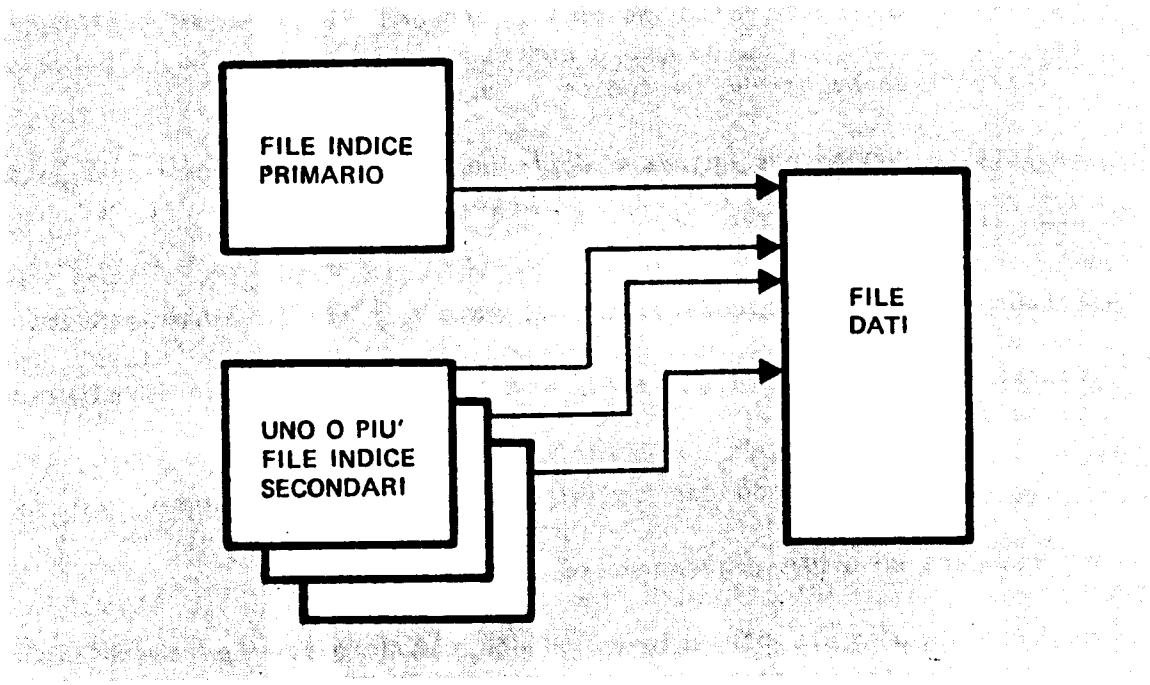


Figura 1-2 Indici secondari

CHIAVI CONCATENATE

A volte può essere necessario accedere ai record usando come chiave la combinazione di più valori. A questo scopo due o più dati vengono combinati (o concatenati) per formare il valore della chiave.

Ad esempio, una società con alcune succursali, potrebbe desiderare di mantenere logicamente separati i clienti delle singole succursali allo scopo di emettere facilmente resoconti per ognuna di esse.

Se la chiave del record è composta dal codice di succursale e dal numero del cliente, allora tutti i clienti di una certa succursale vengono raggruppati e possono essere trattati in modo indipendente rispetto alle altre succursali. Ciò non esclude però la possibilità di una gestione globale di tutto il file clienti.

LETTURA DI RECORD CON CHIAVE

Per leggere un record in un file con chiave si possono utilizzare cinque metodi:

- Lettura random per chiave. In questo metodo si deve fornire un valore di chiave e viene restituito un puntatore al record associato.

  Una volta specificata la chiave, il metodo di accesso esegue una ricerca sul file indice allo scopo di trovare il relativo puntatore al record. Se la ricerca ha avuto successo viene restituito il puntatore, altrimenti viene emesso un codice di errore.

  La lettura random per chiave è utile per accedere ed aggiornare un file quando si voglia un record ben preciso e la chiave del record sia conosciuta.

- Lettura generica per chiave. Questo metodo è simile al precedente con la differenza che, se il record specificato non esiste, viene restituito il puntatore al record con il valore di chiave immediatamente successivo.

  L'uso di questo metodo con chiavi parziali fornisce la possibilità di raggruppare record dati in sezioni logiche ed è un mezzo potente per selezionare ed organizzare record di dati.

- Lettura sequenziale per chiave. Questo metodo offre la possibilità di leggere record in sequenza logica per valori di chiave, indipendentemente dalla sequenza fisica dei record nel file.

  La lettura può cominciare dal primo record logico nel file oppure da qualsiasi altra posizione.

  Questo metodo può essere usato per accedere ai record seguendo una certa sequenza, ad esempio per valore di chiave, oppure insieme alla lettura generica, per leggere gruppi logici di record.

- Lettura sequenziale in ordine fisico. Permette di leggere, dal file dati, i record in ordine fisico, senza riferimenti al file indice.

  Questo metodo è utile per applicazioni di elaborazioni di dati in cui la sequenza di lettura dei record non è importante quanto l'analisi dei dati, e riduce il tempo di accesso ai dati stessi, essendo eliminata la ricerca nel file indice.

- Lettura random per numero di record. Consente di accedere ai dati per mezzo del loro numero di record.

  Ciò permette di trattare un file dati sia in modo DAM che in modo diretto per chiave.

CANCELLAZIONE DI RECORD E DI CHIAVE

Un sistema di gestione di file con chiave offre la possibilità di rimuovere record dati e/o la loro chiave dalla struttura di base. Questo riduce lo sforzo di programmazione richiesto per la gestione dei record cancellati, fa risparmiare spazio su disco e diminuisce la necessità di riorganizzazione dei file.

Esiste un'appropriata funzione di cancellazione che permette di eliminare record da un particolare file. Questa funzione toglie dal file indice la chiave associata al record e rende il record stesso disponibile per riutilizzarlo quando se ne vogliano aggiungere degli altri.

Nel caso si voglia eliminare la chiave associata ad un record conservando il record stesso, occorre usare la funzione di cancellazione di chiave.

Questa funzione rimuove la chiave dal file indice ma non altera il record dati.

Questo record potrà essere letto mediante accesso sequenziale.

La responsabilità di rimozione del record dati é quindi delegata al programmatore.

La cancellazione di chiave è molto utile in applicazioni ove si prevede di accedere al record in tempi successivi come, ad esempio, nella generazione di archivi: è utile anche per rimuovere dai file indice secondari le chiavi dei record cancellati.

ESEMPI DI USO DI FILE CON CHIAVE

La struttura a file con chiave è molto più flessibile di altri tipi di organizzazioni ed è particolarmente adatta per sofisticate applicazioni sia interattive che in tempo reale.

I seguenti esempi illustrano alcune di queste applicazioni:

INTERROGAZIONE DI ARCHIVI

Un campo molto adatto all'uso di file con chiave è l'interrogazione di archivi cioè la possibilità di leggere l'informazione contenuta in un record usando solamente la chiave.

Ad esempio, in un sistema di gestione di magazzino, occorre conoscere la giacenza di un determinato articolo prima di procedere ad un suo eventuale ordine.

Mediante l'uso di un file con chiave, definendo come chiave il codice dell'articolo, un programma può richiedere all'operatore il numero di codice, leggere il record corrispondente e visualizzare la giacenza dell'articolo selezionato.

Sfruttando gli indici secondari si può assegnare una seconda chiave all'articolo (ad esempio la sua descrizione); in questo modo, se non si conosce il numero di codice si può fornire la descrizione dell'articolo per ottenere il record desiderato oppure più record con la stessa descrizione.

CONTROLLI

Un secondo esempio di uso vantaggioso di file con chiave è il controllo di dati. Il valore di un dato può essere usato come chiave in una struttura di file con chiave.

In questo caso il valore fornito dall'operatore può essere verificato mediante una lettura random con chiave sulla struttura del file in cui questo dato è utilizzato come chiave. Se la chiave non esiste, l'operatore è immediatamente informato dell'errore e quindi deve fornire un nuovo valore.

Ad esempio, in un sistema di gestione stipendi, il record delle ore di lavoro settimanali contiene anche la matricola del dipendente.

Il numero di matricola viene impostato dall'operatore ed è usato come chiave per accedere al file del dipendente. Se la chiave non viene trovata, l'operatore ne viene informato e può così impostare il valore corretto.

AGGIORNAMENTI

L'esempio più significativo ed utile di uso di file con chiave è l'aggiornamento in tempo reale dei dati.

Un dato contenuto in un record può essere mantenuto aggiornato rispetto all'evoluzione del programma applicativo; esso diventa così un'informazione significativa in tempo reale.

Normalmente questo avviene congiuntamente a fasi di lettura di record.

Ad esempio, in ambiente bancario, prelievi e depositi possono essere registrati direttamente sui relativi conti correnti per mezzo di una lettura con chiave (n° di conto), aggiungendo o sottraendo la somma trattata e riscrivendo il record nel file dei conti correnti. Ogni lettura o transazione successiva rifletterà la nuova situazione di conto indipendentemente dal tempo trascorso.

INSERIMENTI E CANCELLAZIONI

Record possono essere aggiunti o cancellati in tempo reale dalla struttura di file con chiave. In questo modo sono resi disponibili nuovi dati, mentre quelli non più desiderati possono essere rimossi, liberando spazio per i primi.

Ad esempio, nel sistema di magazzino, un nuovo articolo può diventare immediatamente disponibile per gli ordini mentre un articolo non più in vendita può essere rimosso eliminando così la possibilità di nuovi ordini.

2. STRUTTURA DI FILE ISAM

SOMMARIO

Questo capitolo descrive le strutture ISAM e le relative tecniche di gestione di file.

INDICE

FILE ISAM 2-1

IL FILE DATI 2-1

IL FILE INDICE 2-1

STRUTTURA B TREE 2-2

LA STRUTTURA ISAM AD ALBERO 2-4

BLOCCHI DI CONTROLLO DATI 2-5

NUMERO DI FILE APERTI E FILE BUFFERING 2-6

RECORD CANCELLATI 2-7

STRUTTURA DI FILE ISAM

FILE ISAM

ISAM usa due strutture dati: i file indice ed i file dati. Più file indice possono coesistere se si utilizza una struttura a chiavi secondarie. Questi concetti sono stati descritti nel capitolo 1.

I file dati ISAM sono memorizzati secondo il formato BASIC e sono compatibili con tutte le funzionalità BASIC.

Programmi di utilità come "sort" e "merge" possono quindi essere usati senza alcuna modifica.

Occorre però notare che tali programmi provocano la distruzione dei legami logici tra file dati e file indice.

IL FILE DATI

Il file dati è un file BASIC ad accesso diretto (random) e può quindi essere gestito per mezzo delle istruzioni BASIC relative a tale tipo di file.

Ad esempio, per trasferire un record dal file dati nel relativo buffer verrà usata l'istruzione BASIC GET; analogamente l'istruzione PUT scriverà un record nel file mentre l'istruzione FIELD verrà usata per definire la struttura del record.

ISAM non fornisce alcuna di queste funzioni: esse sono delegate al programmatore.

IL FILE INDICE

Il file indice è formato da record contenenti ognuno un certo numero di chiavi; ad ogni chiave corrisponde un numero di record.

Le chiavi duplicate necessitano di un trattamento speciale, in quanto il meccanismo di ricerca è leggermente diverso. Questo significa che esistono due tipi di file indice: quelli in cui sono permesse chiavi duplicate e quelli che accettano solo chiavi uniche.

In ogni caso le chiavi non sono memorizzate in sequenza fisica ma sono collegate mediante puntatori che le mantengono in ordine alfanumerico.

Questa sequenza è gestita per mezzo di una struttura "ad albero".

La struttura ad albero usata da ISAM é detta "B+tree".

STRUTTURA B TREE

Prima di entrare in dettaglio nella struttura B+tree usata da ISAM conviene esaminare una forma più semplice allo scopo di comprendere meglio il suo funzionamento.

In una particolare forma di B tree, ogni chiave occupa un nodo dell'albero ed ha asssociati due puntatori: uno alla chiave immediatamente precedente ed uno a quella seguente; queste due chiavi giacciono nel livello immediatamente sottostante.

Questa struttura è nota anche con il nome di "albero binario".

La Figura 2-1 illustra la struttura B tree dopo l'aggiunta delle chiavi "4.5" e "9" alla chiave "7" pre-esistente.

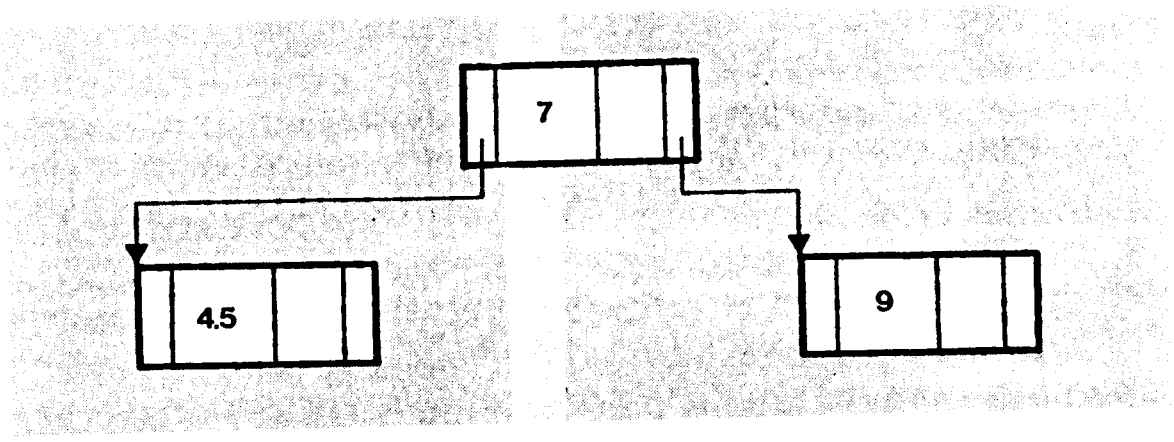


Figura 2-1 Struttura B tree 1

Ogniqualvolta un record viene aggiunto al file, la sua chiave é confrontata con il valore della chiave che è in cima all'albero; questo valore è detto "radice" dell'albero. Se la chiave del record da inserire è maggiore di quella della radice, allora essa viene confrontata con quella associata al puntatore alla destra della radice; analogamente, se la chiave del record da inserire è minore di quella della radice, essa viene confrontata con il nodo puntato a sinistra. Successivi confronti vengono effettuati fino a che si raggiunge la corretta posizione della chiave nel livello più basso dell'albero.

Ad esempio, aggiungendo le chiavi 2, A3, A5, 4.5A e 9.9 alla struttura di Figura 2-1, otterremo la struttura della Figura 2-2.

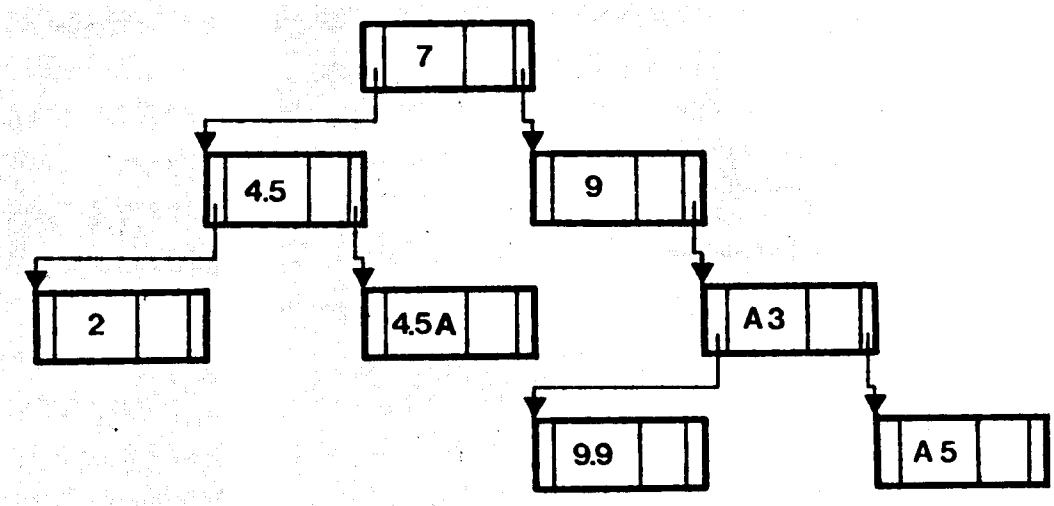


Figura 2-2 Struttura B tree II

Fintantoché l'albero rimane bilanciato, cioè il numero di chiavi alla sinistra di un nodo è paragonabile al numero di quelle alla destra, il tempo di ricerca di una chiave è ottimizzato.

Ad esempio, la chiave A5 della figura 2-2 può essere raggiunta dopo aver esaminato i nodi 7, 9 ed A3, cioè dopo tre confronti.

Nel caso in cui le chiavi vengano inserite in sequenza alfanumerica allora avremmo la struttura di Figura 2-3 nella quale sono necessari sette confronti per raggiungere la suddetta chiave.

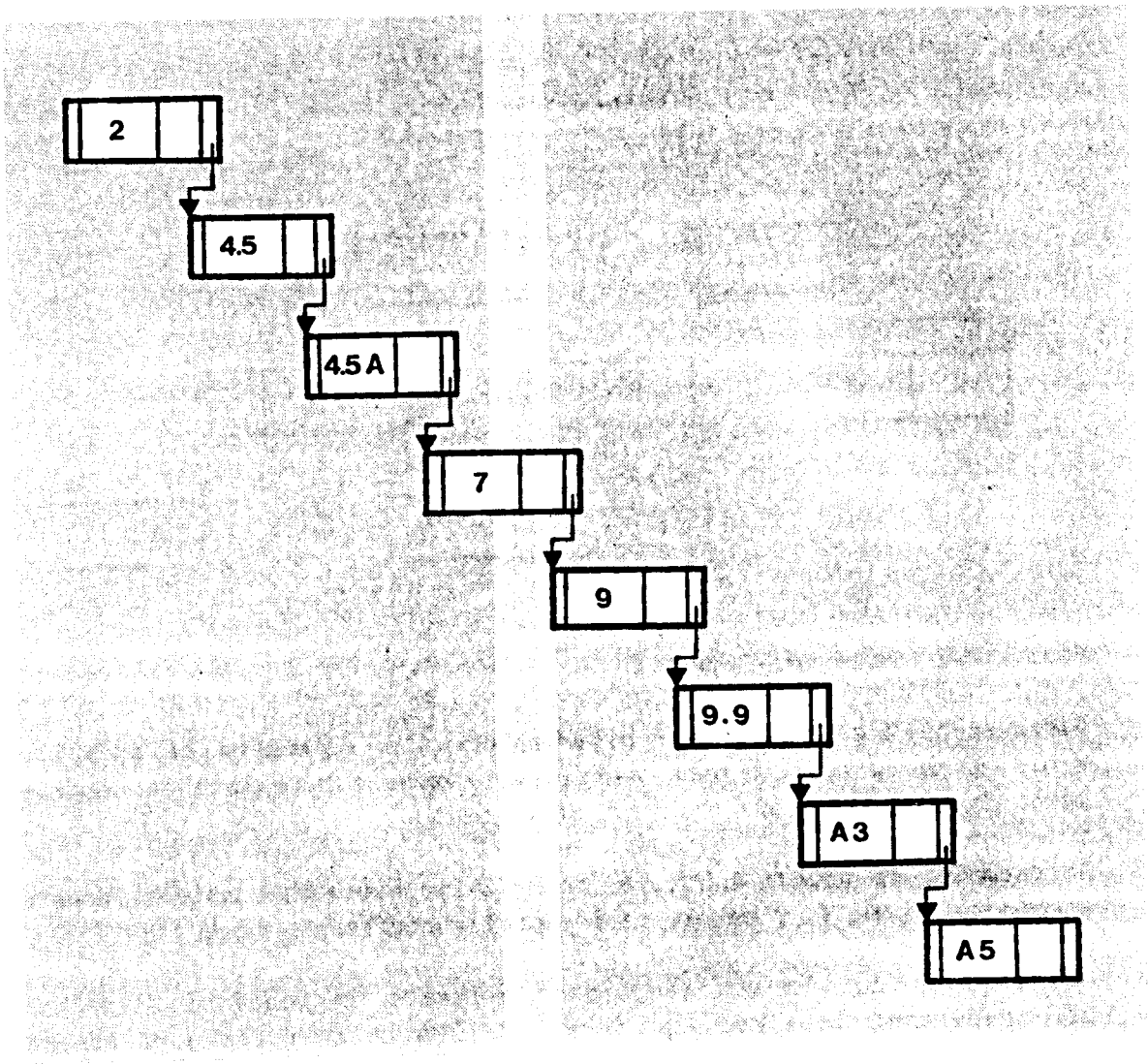


Figura 2-3 Struttura B tree III

Allo scopo di evitare sbilanciamenti di questo genere occorre disporre di un algoritmo il quale, mediante aggiustamenti ("splitting") all'albero, lo mantenga bilanciato.

LA STRUTTURA ISAM AD ALBERO

Il B+tree usato da ISAM non è semplice come quello appena descritto. ISAM usa due tipi di record: i nodi indice ed i nodi terminali.

I nodi terminali risiedono nel livello più basso dell'albero e contengono tutte le chiavi con i corrispondenti numeri del record dati.

I nodi indice occupano invece tutti i livelli dell'albero, a parte quello inferiore, ed offrono un cammino per l'accesso ai nodi terminali.

STRUTTURA DI FILE ISAM

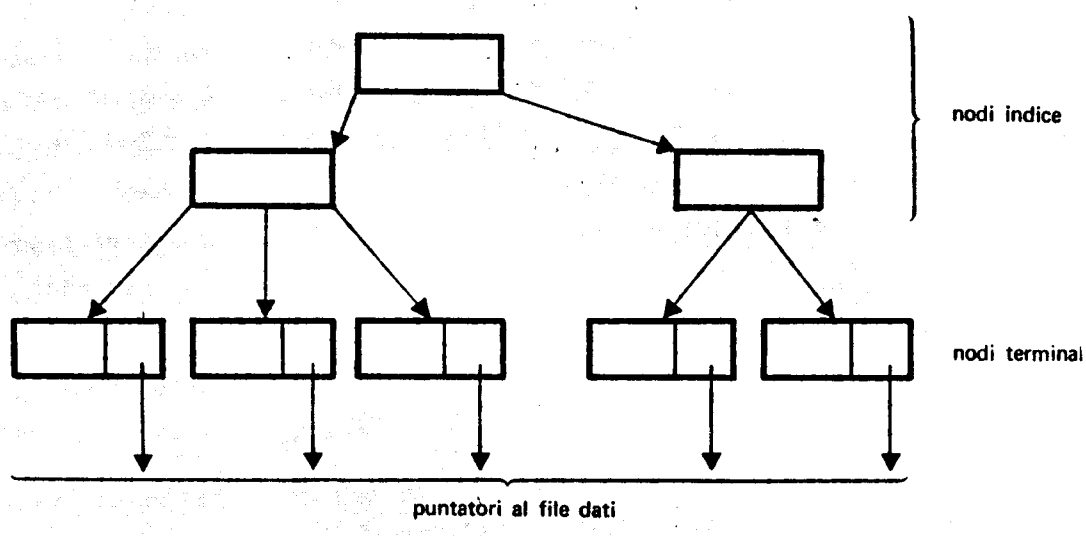


Figura 2-4 Struttura dati ISAM

Inoltre, ogni nodo terminale possiede due puntatori, uno al nodo terminale precedente ed uno al nodo terminale seguente. Questo permette di accedere a tutte le chiavi, rispettando la sequenza ASCII.

Dato che la lunghezza del record nel file indice è fissa mentre la lunghezza delle chiavi è variabile, anche il numero delle chiavi per record risulta essere variabile. Un esempio di tale struttura é illustrato nell'appendice B.

BLOCCHI DI CONTROLLO DATI

Un blocco di controllo dati (DCB), deve venir allocato ogni volta che un file indice viene aperto. Questo blocco contiene informazioni sull'uso corrente del file. Una di queste è il puntatore al blocco "header" (HCB, Header Control Block) che è un record contenuto nel file indice stesso e contiene a sua volta l'informazione di header. Questa informazione viene usata per una corretta lettura del dato relativo alla chiave successiva, oltreché per accedere alla lista dei record cancellati e poter operare sul primo di essi ed infine per accedere al successivo nodo disponibile per la scrittura sequenziale.

Il DCB contiene anche un flag che indica se il file é di tipo unico o se può contenere chiavi duplicate.

NUMERO DI FILE APERTI E FILE BUFFERING

L'utente deve definire il numero massimo di file indice che possono essere aperti contemporaneamente; esiste una primitiva in ISAM che permette di specificare questo numero che deve essere compreso tra 1 e 15.

I numeri associati ai file indice usati da ISAM devono essere diversi da quelli assegnati ai file dati; ad esempio, se ISAM è stato inizializzato per un massimo di quattro file, siano 1,2,3 e 4, i numeri disponibili per file diversi da file indice sono dal 5 al 15.

Se un programma BASIC fa uso dei numeri 1, 2, 3 o 4, si possono verificare degli errori imprevedibili. Un metodo per evitare conflitti del genere è quello di numerare i file dati in ordine decrescente, iniziando dal più alto numero di file disponibile.

I record del file indice vengono letti tramite "buffers" di file (aree temporanee) allocate nella memoria del sistema.

Il sistema riserva automaticamente un minimo di due buffer per ogni file, permettendo il funzionamento dell'algoritmo di bilanciamento del B+tree nel file indice.

Tuttavia l'utente ha la possibilità di aumentare il numero di buffer che è limitato solamente dalla disponibilità di memoria: mano a mano che cresce il numero di file, cresce pure la quantità di memoria necessaria alla loro gestione (cioé DCB, ecc.) e decresce quindi la quantità di memoria disponibile per i buffer.

E' consigliabile quindi minimizzare il numero di file aperti lasciando disponibile più memoria possibile per i buffer.

Anche se sono consigliati due buffer per file indice, ISAM funziona normalmente anche in presenza di un solo buffer; in questo caso però la sua efficienza diminuisce.

ISAM è in grado di fare ciò perché contiene algoritmi che allocano i buffer disponibili per i file che sono stati usati più recentemente, facendo così il miglior uso possibile dello spazio del buffer.

Cioé, quando un file indice richiede un buffer addizionale e tutti i buffer sono utilizzati contemporaneamente, viene allocato il buffer che è stato usato per ultimo.

RECORD CANCELLATI

La cancellazione di un record è implementata nel modo seguente: la sua chiave viene rimossa dal file indice ed il relativo numero di record viene salvato per poter essere riutilizzato in successive operazioni di scrittura.

I record cancellati vengono salvati in uno "stack L.I.F.O." (Last In First Out) in modo che l'ultimo tra quelli cancellati sarà il primo ad essere riutilizzato.

3. USO DI ISAM IN UN PROGRAMMA BASIC

SOMMARIO

Questo capitolo descrive come comunicare con la subroutine ISAM e come usare le singole funzioni all'interno di un programma BASIC.

INDICE

FILE PROGRAMMA ISAM 3-1

USO DI ISAM SU M20 3-1

USO DI ISAM IN UN PROGRAMMA BASIC 3-1

COMUNICAZIONE CON ISAM 3-2

CODICI DI RITORNO 3-8

FUNZIONI DI UTILITA' ISAM 3-10

FUNZIONE DI STATO (MS) 3-10

FUNZIONE DI INIZIALIZZAZIONE (MI) 3-11

APERTURA E CHIUSURA DI FILE ISAM 3-13

APERTURA DI FILE (OO) 3-13

CREAZIONE DI FILE (OC) 3-14

APERTURA/CREAZIONE DI FILE (OF) 3-15

CHIUSURA DI FILE (CL) 3-16

LETTURA DATI DA UN FILE ISAM 3-18

LETTURA CON CHIAVE (RK, SK) 3-18

LETTURA GENERICA (RG, SG) 3-20

LETTURA SEQUENZIALE (RN, SN) 3-23

LETTURA SEQUENZIALE ALL'INDIETRO (RP, SP) 3-25

INSERIMENTO DI DATI IN UN FILE ISAM 3-27

INSERIMENTO ADD (WA, SA) 3-27

FUNZIONI DI CANCELLAZIONE 3-31

CANCELLAZIONE DI CHIAVE (KD, SD) 3-31

CANCELLAZIONE DI RECORD (DR) 3-33

USO DI ISAM IN UN PROGRAMMA BASIC

FILE PROGRAMMA ISAM

ISAM viene fornito su un disco a parte. Su questo disco vi sono quattro file che riguardano ISAM:

- isamx.bas
- isamd.bas
- isam.bas
- isam.sav

isamx.bas è un programma didattico sui concetti ISAM; è descritto nell'Appendice A.

isamd.bas è un programma di utilità che permette di esaminare le informazioni contenute in un file; è descritto nell'Appendice B.

isam.bas è il file che contiene l'interfaccia BASIC alla subroutine ISAM.

isam.sav contiene la subroutine ISAM.

USO DI ISAM SU M20

Accendere l'M20, inserire il disco di sistema e premere CR.

Dopo alcuni secondi compariranno sul video alcune informazioni di sistema seguite dal simbolo ">".

A questo punto si può richiamare l'interprete BASIC per mezzo del comando BA, e quindi scrivere il proprio programma BASIC.

USO DI ISAM IN UN PROGRAMMA BASIC

ISAM è una subroutine del programma utente BASIC. Essa viene richiamata dall'istruzione GOSUB 6ØØØØ.

Un programma BASIC comunica con ISAM per mezzo di variabili che specificano la struttura del file da usare, la funzione da eseguire e la chiave del record.

Un programma BASIC che usa ISAM è soggetto alle seguenti limitazioni:

- deve includere l'istruzione che inserisce la subroutine ISAM nel programma stesso, occorre cioè aggiungere i seguenti comandi:

MERGE "isam.bas"	Inserisce la subroutine ISAM
SAVE "1:MYPROG"	Registra (sul disco inserito nell'unità uno) il programma utente con la subroutine ISAM nel file MYPROG.

- il programma chiamante non deve contenere istruzioni numerate nell'intervallo 6ØØØØ-6Ø11Ø altrimenti possono verificarsi errori.

- vengono utilizzati due vettori denominati:

  ISAM$ e ISAM%

  Questi vettori devono essere usati solamente per comunicare con ISAM. Non é necessario dimensionarli in quanto si utilizzano solo gli elementi con indice da 1 a 9.

Tutti i programmi che usano ISAM devono eseguire le seguenti operazioni:

PASSO	OPERAZIONE
1	Aprire il file indice ed il file dati
2	Assegnare i valori alle variabili da passare ad ISAM
3	Richiamare la subroutine ISAM
4	Controllare il codice di ritorno per vedere se l'operazione è terminata correttamente
5	Leggere, scrivere o cancellare record dati, come richiesto
6	Chiudere tutti i file

COMUNICAZIONE CON ISAM

Tutte le comunicazioni con ISAM avvengono per mezzo dei due vettori ISAM$ e ISAM%. Le tavole seguenti descrivono la loro struttura:

VARIABILE NUMERICA	SIGNIFICATO
ISAM% (1)	Numero del DCB del file. E' il numero del DCB usato per gestire tutte le operazioni sul file. E' usato da tutte le funzioni per indicare quale è, tra quelli aperti, il file indice da usare. Il suo valore deve essere compreso tra 1 e 15.
ISAM% (2)	Non usato
ISAM% (3)	Non usato
ISAM% (4)	Non usato
ISAM% (5)	Numero del record dati. Questa variabile è usata per la costruzione di un indice secondario per record già esistenti. E' usata dalla scrittura Add secondaria (SA) e deve essere specificata per tutti gli accessi con chiave su file indici aventi chiavi duplicate.
ISAM% (6)	Numero di buffer di disco addizionali Questa variabile è usata dalla funzione di inizializzazione (MI) per specificare il numero massimo di buffer da allocare per ogni file, in aggiunta ai due allocati dal sistema.
ISAM% (7)	Numero massimo di file aperti. Questa variabile è usata dalla funzione di inizializzazione (MI) per specificare il numero massimo di file indice che possono essere aperti contemporaneamente. Il suo valore deve essere compreso tra 1 e 15.
ISAM% (8)	Non usato
ISAM% (9)	Non usato

Tabella 3-1 Variabili numeriche passate a ISAM

VARIABILE STRINGA	SIGNIFICATO
ISAM$ (1)	Codice della funzione. Deve essere specificato per tutte le funzioni dato che indica la funzione ISAM da eseguire: Valori corretti: RK - Lettura con chiave RG - Lettura generica RN - Lettura in avanti RP - Lettura all'indietro WA - Scrittura Add DR - Cancellazione di record KD - Cancellazione di chiave SK - Lettura con chiave secondaria SG - Lettura generica secondaria SN - Lettura in avanti secondaria SP - Lettura all'indietro secondaria SA - Scrittura Add secondaria SD - Cancellazione di chiave secondaria OO - Apertura di file OC - Creazione di file OF - Apertura/ creazione di file CL - Chiusura di file MI - Funzione di inizializzazione MS - Funzione di stato
ISAM$ (2)	Valore della chiave. E' richiesto dalle seguenti funzioni: RK/SK RG/SG WA/SA DR KD/SD La lunghezza della chiave può essere al massimo 11Ø caratteri ASCII.
ISAM$ (3)	Non usato

Tabella 3-2 Variabili stringa passate a ISAM (cont.)

VARIABILE STRINGA	SIGNIFICATO
ISAM$ (4)	Nome del file indice. Specifica il nome del file indice. E' usato dalle funzioni di apertura, creazione ed apertura/creazione di file. Il nome del file deve essere conforme allo standard PCOS.
ISAM$ (5)	Non usato
ISAM$ (6)	Flag di chiave duplicata/unica Viene usato durante la creazione di un nuovo file indice per specificare sè le chiavi duplicate sono ammesse o meno. Il valore è "D" per chiave duplicata ed "U" per chiave unica. Un valore nullo è considerato come "U".
ISAM$ (7)	Non usato
ISAM$ (8)	Non usato
ISAM$ (9)	Non usato

Tabella 3-2 Variabili stringa passate a ISAM

Le tavole seguenti descrivono l'uso dei vettori ISAM% e ISAM$ che restituiscono variabili di ISAM al programma chiamante:

VARIABILE NUMERICA	SIGNIFICATO
ISAM% (1)	Non usato
ISAM% (2)	Non usato
ISAM% (3)	Non usato
ISAM% (4)	Non usato

Tabella 3-3 Variabili numeriche restituite a ISAM (cont.)

VARIABILE NUMERICA	SIGNIFICATO
ISAM% (5)	Non usato
ISAM% (6)	Numero di record attivi. Questo valore è ritornato dalla funzione di stato (MS) ed indica il numero di chiavi (e quindi di record) attivi nel sistema.
ISAM% (7)	Numero di record cancellati non usati. Questo valore è restituito dalla funzione di stato (MS) ed indica il numero di record cancellati che non sono più stati usati da successive funzioni di scrittura.
ISAM% (8)	Codice di ritorno. Può assumere i seguenti valori: ØØ - Funzione eseguita correttamente 31 - Sequenza funzionale errata 41 - Errore di sintassi 51 - Record non trovato 61 - Chiave duplicata 71 - Errore di apertura/chiusura 81 - Errore di disco Una descrizione dettagliata di ogni errore è data nella tabella 3-5.
ISAM% (9)	Numero del record dati. Questo valore viene restituito dopo ogni operazione conclusa correttamente e può essere usato in seguito dalle istruzioni I/O del BASIC per leggere o scrivere dati.

Tabella 3-3 Variabili numeriche restituite da ISAM

VARIABILE STRINGA	SIGNIFICATO
ISAM$ (1)	Non usato
ISAM$ (2)	Valore della chiave. Questo valore è restituito dalle funzioni di lettura (eccetto per la funzione di lettura con chiave): RG/SG RN/SN RP/SP
ISAM$ (3)	Non usato
ISAM$ (4)	Non usato
ISAM$ (5)	Non usato
ISAM$ (6)	Tipo duplicato/unico. Questo valore è restituito dalla funzione di stato (MS) per indicare il tipo di file.
ISAM$ (7)	Non usato
ISAM$ (8)	Non usato
ISAM$ (9)	Non usato

Tabella 3-4 Variabili stringa restituite da ISAM

CODICI DI RITORNO

La tavola seguente contiene i valori dei possibili codici di ritorno:

NUMERO DI CODICE	SIGNIFICATO	FUNZIONI PERMESSE
ØØ	Ritorno normale. Indica che la funzione richiesta è stata eseguita correttamente e che il numero di record restituito è valido.	Tutte
31	Sequenza funzionale errata. Indica che la sequenza di chiamata delle funzioni non è corretta. Questo succede, ad esempio, quando la prima chiamata non è stata una funzione di inizializzazione (MI).	Tutte
41	Errore di sintassi Indica che alcuni valori errati sono stati passati alla funzione	Tutte
51	Non trovato il record. Indica che il record non è stato trovato per una delle seguenti ragioni: - La chiave non esiste - Il valore della chiave è maggiore del valore di chiave più alto presente nel file. - Si è tentato di leggere oltre la fine del file (o prima dell'inizio) In tutti e tre i casi il numero del record restituito è errato	 RK/SK RG/SG RN/SN RP/SP

Tabella 3-5 Codici di ritorno (cont.)

NUMERO DI CODICE	SIGNIFICATO	FUNZIONI PERMESSE
61	Chiave duplicata Indica che il record da aggiungere possiede la stessa chiave di un record già esistente nel file. Questo codice viene restituito nel caso di file di tipo a chiave unica, oppure nel caso di file a chiave duplicata se esistono già la chiave specificata ed il numero di record.	WA/SA
71	Errore di apertura/ chiusura. Indica che la relativa funzione non è stata eseguita correttamente per una delle ragioni seguenti: - Il DCB indicato è già in uso - Il file non esiste - Il DCB non è stato aperto	 00 0C 0F 00 tutte eccetto 00 0C 0F
81	Errore di disco. Indica che un errore di disco si è verificato e che ISAM non può effettuare alcun recupero (recovery). Cause probabili sono: - La directory è completa - Non c'è più spazio libero sul disco	 0C 0F WA/SA DR

Tabella 3-5 Codici di ritorno

FUNZIONI DI UTILITA' ISAM

FUNZIONE DI STATO (MS)

La funzione di stato (MS) restituisce informazioni sullo stato attuale di un file. Questa informazione comprende il numero di record dati attivi e il numero di record cancellati e non più riutilizzati.

Le seguenti variabili devono essere passate a ISAM.

ISAM% (1) - numero di DCB del file
ISAM$ (1) - codice di funzione (MS)

Le seguenti variabili sono restituite:

ISAM% (6) - numero di record dati attivi
ISAM% (7) - numero di record cancellati e non più usati
ISAM% (8) - codice di ritorno
ISAM$ (6) - flag di chiave duplicata/unica

Per avere informazioni su un file mediante la funzione MS occorre seguire la seguente sequenza:

PASSO	OPERAZIONE
1	Aprire il file dati mediante l'istruzione OPEN del BASIC
2	Strutturare il buffer del file mediante l'istruzione FIELD del BASIC
3	Aprire il file indice mediante la funzione Open di ISAM
4	Specificare il numero di DCB del file
5	Porre il codice di funzione ad "MS"
6	Richiamare la subroutine ISAM
7	Controllare il codice di ritorno
8	Chiudere il file dati mediante l'istruzione CLOSE del BASIC
9	Chiudere il file indice mediante l'appropriata funzione ISAM

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ OPEN "R",15,"DFILE",128 11Ø FIELD 15,128 AS A$ 12Ø ISAM%(1) = 1 13Ø ISAM$(1) ="00" 14Ø ISAM$(4) ="INDEX" 15Ø GOSUB 6ØØØØ 16Ø IF ISAM%(8)<>Ø THEN GOTO 9ØØ	Le istruzioni 1ØØ e 11Ø aprono il file dati DFILE e ne strutturano il buffer. Le istruzioni da 12Ø a 16Ø aprono il file indice INDEX ed assegnano il valore 1 al relativo DCB.
. . . 2ØØ ISAM%(1) = 1 21Ø ISAM$(1) = "MS" 22Ø GOSUB 6ØØØØ 23Ø IF ISAM%(8)<>Ø THEN GOTO 1ØØØ	Le istruzioni da 2ØØ a 23Ø eseguono la funzione MS di ISAM applicata al file indice
. . . 5ØØ CLOSE 15 51Ø ISAM%(1) = 1 52Ø ISAM$(1) = "CL" 53Ø GOSUB 6ØØØØ 54Ø IF ISAM%(8)<>Ø THEN GOTO 11ØØ	L'istruzione 5ØØ chiude il file dati. La sequenza da 51Ø a 54Ø chiude il relativo file indice.

FUNZIONE DI INIZIALIZZAZIONE (MI)

La funzione di inizializzazione (MI) deve essere la prima funzione ISAM richiamata dal programma utente. Il programma deve specificare il numero di buffer addizionali oltre ai due assegnati automaticamente dal sistema.

Occorre ricordare però che troppi buffer allocati possono rendere insufficiente la memoria utente.

Il programma deve specificare anche il numero massimo di file indice che possono essere aperti contemporaneamente. Questo numero deve essere compreso fra 1 e 15 e deve essere mantenuto al valore minimo per lasciare più memoria ai buffer di sistema.

I numeri dei file usati da ISAM non devono essere usati dal programma BASIC, in caso contrario si potrà verificare un errore non previsto.

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(6) - numero di buffer addizionali
ISAM%(7) - numero massimo di file indice aperti
ISAM$(1) - codice di funzione (MI)

La seguente variabile è ritornata:

ISAM%(8) - codice di ritorno

I valori possibili del codice di ritorno sono: 31.

Per inizializzare ISAM il programma deve seguire la sequenza indicata di seguito:

PASSO	OPERAZIONE
1	Specificare il numero di buffer addizionali
2	Specificare il numero massimo di file aperti
3	Porre il codice di funzione ad "MI"
4	Richiamare la subroutine ISAM
5	Controllare il codice di ritorno

Esempio

PROGRAMMA BASIC	COMMENTI
2ØØ ISAM%(6) =2 21Ø ISAM%(7) =2 22Ø ISAM$(1) ="MI" 23Ø GOSUB 6ØØØØ 24Ø IF ISAM%(8)<>Ø THEN GOTO 9ØØ	Le istruzioni da 2ØØ a 24Ø inizializzano ISAM allocando due buffer addizionali e ponendo a due il numero massimo di file che possono essere aperti contemporaneamente. I numeri dei file 1 e 2 non devono essere usati dal programma BASIC.

Pagina mancante?

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ ISAM%(1)=2 11Ø ISAM$(1)="00" 12Ø ISAM$(4)="IFILE" 13Ø GOSUB 6ØØØØ 14Ø IF ISAM%(8)<>Ø THEN GOTO 6ØØ . . .	Le istruzioni da 1ØØ a 14Ø provvedono all'apertura del file indice IFILE, presupponendo che questo sia già stato creato, ed assegnano il valore 2 al DCB.

CREAZIONE DI FILE (OC)

La funzione di creazione di file provvede alla creazione e all'apertura di un file indice. Occorre specificare il nome del file ed il numero di DCB.

Se esiste già un file con il nome specificato, allora il file non verrà creato e verrà restituito il codice 71.

Il file creato può contenere solo chiavi uniche o, se possibile, chiavi duplicate: la scelta è delegata al programma chiamante il quale può assegnare alla variabile ISAM$ (6) i valori "D" (per duplicata) oppure "U" (per unica). Il valore di default è "U".

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - numero di DCB del file
ISAM$(1) - codice di funzione (OC)
ISAM$(4) - nome del file indice
ISAM$(6) - chiave duplicata "D" o chiave unica "U".

La seguente variabile è restituita al programma:

ISAM%(1) - numero di DCB del file
ISAM%(8) - codice di ritorno

I valori possibili del codice di ritorno sono: ØØ, 31, 41, 71 e 81.

Per creare un file indice usando la funzione di creazione di un file il programma deve rispettare la sequenza:

PASSO	OPERAZIONE
1	Specificare il numero del DCB
2	Porre il codice di funzione a "OC"
3	Specificare il nome del file
4	Richiamare la subroutine ISAM
5	Controllare il codice di ritorno

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ ISAM%(1)=4 11Ø ISAM$(1)="OC" 12Ø ISAM$(4)="IFILE1" 13Ø ISAM$(6)="D" 14Ø GOSUB 6ØØØØ 15Ø IF ISAM%(8)<>Ø THEN GOTO 6ØØ . . .	Le istruzioni da 1ØØ a 15Ø creano ed aprono un file indice con il nome IFILE1, specificando chiavi duplicate ed assegnando DCB 4 al file.

APERTURA/CREAZIONE DI FILE (OF)

Questa funzione apre un file indice già esistente oppure lo crea e lo apre se non esiste.

Riguardo la creazione di un file, è possibile scegliere tra chiavi uniche e chiavi duplicate: si hanno chiavi duplicate ponendo ISAM$(6)="D" oppure chiavi uniche ponendo ISAM$(6)="U". Il valore di default è "U".

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - numero del DCB del file
ISAM$(1) - codice di funzione (OF)
ISAM$(4) - nome del file indice
ISAM$(6) - chiavi uniche "U" o duplicate "D".

Viene restituita la seguente variabile:

ISAM%(8) - codice di ritorno

Possibili valori del codice di ritorno sono: ØØ, 31, 41, 71 e 81.

Per aprire/creare un file con questa funzione, un programma deve seguire la sequenza:

PASSO	OPERAZIONE
1	Specificare il DCB
2	Porre il codice funzione a "OF"
3	Specificare il nome del file
4	Richiamare la subroutine ISAM
5	Controllare il codice di ritorno

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ ISAM%(1)=3 11Ø ISAM$(1)="OF" 12Ø ISAM$(4)="IFILE" 13Ø ISAM$(6)="U" 14Ø GOSUB 6ØØØØ 15Ø IF ISAM%(8)<>Ø THEN GOTO 6ØØ	Le istruzioni da 1ØØ A 15Ø creano ed aprono un file indice con il nome IFILE il quale può contenere solo chiavi uniche. L'istruzione 1ØØ assegna il valore 3 al DCB.

CHIUSURA DI FILE (CL)

La funzione di chiusura file chiude un file indice aperto identificato da un particolare DCB.

Se il valore di DCB non corrisponde ad un file aperto, viene restituito il codice 71.

USO DI ISAM IN UN PROGRAMMA BASIC

Le seguenti variabili devono essere comunicate ad ISAM:

ISAM%(1) - Numero di DCB del file
ISAM$(1) - Codice di funzione (CL)

Viene restituita la seguente variabile:

ISAM%(8) - Codice di ritorno

Possibili valori del codice di ritorno sono: ØØ, 31, 41 e71

Per chiudere un file indice usando la funzione "Close File", il programma deve seguire la sequenza:

PASSO	OPERAZIONE
1	Specificare il DCB del file
2	Porre il codice di funzione a "CL"
3	Richiamare la subroutine ISAM
4	Controllare il codice di ritorno

Esempio

PROGRAMMA BASIC	COMMENTI
2ØØ ISAM%(1)=1 21Ø ISAM$(1)="CL" 22Ø GOSUB 6ØØØØ 23Ø IF ISAM%(8)<>Ø THEN GOTO 6ØØ	Le istruzioni da 2ØØ a 3ØØ chiudono il file identificato da DCB 1.

LETTURA DATI DA UN FILE ISAM

LETTURA CON CHIAVE (RK, SK)

La funzione di lettura con chiave cerca la chiave fornita nel file indice e ritorna il numero del record dati associato.

Se la chiave non viene trovata viene ritornato il codice 51.

Le funzioni RK e SK sono funzionalmente identiche.

Quando si effettua una lettura con chiave su un file indice a chiavi duplicate, occorre anche specificare il numero del record cosicché il valore della chiave ed il numero del record possono sempre essere usati per allocare la chiave fornita.

Se il numero del record è Ø allora la funzione "Read Key" restituisce il numero del record associato al primo valore della chiave della lista dei duplicati.

I seguenti record della lista potranno essere letti mediante la funzione di lettura in avanti ("Read Next").

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - Numero del DCB del file
ISAM%(5) - Numero del record dati (solo per chiavi duplicate)
ISAM$(1) - Codice di funzione (RK o SK)
ISAM$(2) - Valore della chiave

Vengono restituite le seguenti variabili:

ISAM%(8) - Codice di ritorno
ISAM%(9) - Numero del record dati

I valori possibili di ritorno sono: ØØ, 31, 41, 51 e 71.

Per leggere un record dal file dati usando la funzione "Read Key", sono necessari i seguenti passi:

USO DI ISAM IN UN PROGRAMMA BASIC

PASSO	OPERAZIONE
1	Aprire il file dati mediante l'istruzione OPEN del BASIC
2	Strutturare il buffer del file mediante l'istruzione FIELD del BASIC
3	Aprire il file indice mediante la funzione di apertura di file ("Open File")
4	Specificare il numero del DCB
5	Porre il codice funzione a "RK" o "SK"
6	Specificare il valore della chiave
7	Specificare il numero del record (solo per file di tipo a chiavi duplicate).
8	Richiamare la subroutine ISAM
9	Controllare il codice di ritorno
10	Leggere il record trovato mediante istruzione GET del BASIC
11	Chiudere il file dati con l'istruzione CLOSE del BASIC
12	Chiudere il file indice mediante l'appropriata funzione di chiusura di file ("Close File").

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ OPEN "R",15,"1:DATAFILE",128 11Ø FIELD 15,128 AS A$ 12Ø ISAM%(1)=1 13Ø ISAM$(1)="00" 14Ø ISAM$(4)="INDEXFILE" 15Ø GOSUB 6ØØØØ 16Ø IF ISAM%(8)<>Ø THEN GOTO 6ØØ : :	Le istruzioni 1ØØ e 11Ø aprono il file dati random sul disco montato sull'unità 1 e strutturano il buffer. Le istruzioni da 12Ø a 16Ø aprono il file indice INDEXFILE e gli assegnano DCB 1.

PROGRAMMA BASIC	COMMENTI
2ØØ ISAM%(1)=1 21Ø ISAM$(1)="RK" 22Ø ISAM$(2)="SMITH" 225 ISAM%(5)=DATAREC% 23Ø GOSUB 6ØØØØ 24Ø IF ISAM%(8)<>Ø THEN GOTO 5ØØ 25Ø GET 15, ISAM%(9) . . . 9ØØ CLOSE 15 91Ø ISAM%(1)=1 92Ø ISAM$(1)="CL" 93Ø GOSUB 6ØØØØ 94Ø IF ISAM%(8)<>Ø THEN GOTO 99Ø . . .	Le istruzioni da 2ØØ a 25Ø leggono il record dati la cui chiave è "SMITH". L'istruzione 225 presuppone un file a chiave duplicata. Viene letto il numero del record dati da una lista variabile. L'istruzione 9ØØ chiude il file dati. Le istruzioni da 91Ø a 94Ø chiudono il file indice.

LETTURA GENERICA (RG/ SG)

La funzione di lettura generica ("Read Generic") restituisce il record dati la cui chiave coincide o è immediatamente superiore a quella specificata.

Questa funzione è utile per accedere al primo record di un gruppo logicamente correlato di record ed avere così un punto di partenza per la lettura sequenziale.

Le funzioni RG e SG sono funzionalmente identiche.

Nel caso di trattamento di un gruppo di record, è responsabilità del programmatore identificarne la fine mediante controlli sul cambiamento del valore dell'indicatore di quel gruppo.

Nel caso di chiavi duplicate, occorre specificare anche il numero di record. Il valore della chiave, congiuntamente a questo numero, è usato per identificare univocamente la chiave specifica.

Assegnando il valore zero al numero di record relativo, la funzione di lettura generica ("Read Generic") restituirà il numero del record corrispondente alla prima chiave trovata nell'indice; gli altri record

aventi la stessa chiave potranno essere letti usando la funzione di lettura in avanti ("Read Next").

ISAM ritorna il numero del record in ISAM%(9); il contenuto del record potrà quindi essere letto mediante l'istruzione GET del BASIC.

La chiave del record trovato viene restituita in ISAM$(2).

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - Numero del DCB del file
ISAM%(5) - Numero di record relativo (solo per chiavi duplicate)

ISAM$(1) - Codice di funzione (RG o SG)
ISAM$(2) - Valore della chiave generica

Le seguenti variabili sono restituite al programma:

ISAM%(8) - Codice di ritorno
ISAM%(9) - Numero del record dati
ISAM$(2) - Valore della chiave trovata

I codici di ritorno possibili sono: ØØ, 31, 41 e 51.

Per leggere un record dal file dati mediante la funzione di lettura generica ("Read Generic") si deve rispettare la sequenza:

PASSO	OPERAZIONE
1	Aprire i file dati mediante l'istruzione OPEN del BASIC
2	Strutturare il buffer del file dati in campi, mediante l'istruzione FIELD del BASIC
3	Aprire il file indice mediante la funzione di apertura file ("Open File")
4	Specificare il numero del DCB del file
5	Porre il codice di funzione ad "RG" o "SG"
6	Specificare il numero del record dati (solo per file di tipo a chiavi duplicate)
7	Richiamare la subroutine ISAM

PASSO	OPERAZIONE
8	Controllare il codice di ritorno
9	Leggere il contenuto del record usando l'istruzione GET del BASIC
10	Chiudere il file dati usando l'istruzione CLOSE del BASIC
11	Chiudere il file indice con la funzione di chiusura di file ("Close File").

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ OPEN "R",14,"DFILE",128 11Ø FIELD 14,128 AS A$ 12Ø ISAM%(1)=4 13Ø ISAM$(1)="00" 14Ø ISAM$(4)="IFILE1" 15Ø GOSUB 6ØØØØ 16Ø IF ISAM%(8)<>Ø THEN GOTO 7ØØ . . .	Le istruzioni 1ØØ e 11Ø aprono il file dati DFILE e strutturano il buffer dati. Le istruzioni da 12Ø a 15Ø aprono il file indice IFILE1 assegnando il valore 4 al relativo DCB.
2ØØ ISAM%(1)=4 21Ø ISAM$(1)="SG" 22Ø ISAM$(2)="1ØØØ" 225 ISAM%(5)=DATAREC% 23Ø GOSUB 6ØØØØ 24Ø IF ISAM%(8)<>Ø THEN GOTO 5ØØ 25Ø GET 14,ISAM%(9)	Le istruzioni da 2ØØ a 25Ø effettuano una lettura generica ("Read Generic") sul file dati usando il file indice secondario IFILE1. L'istruzione 225 è necessaria solo nel caso che IFILE1 contenga chiavi duplicate. L'istruzione 25Ø legge un record la cui chiave coincide o è immediatamente superiore a 1ØØØ. La chiave letta viene assegnata ad ISAM$(2).

PROGRAMMA BASIC	COMMENTI
9ØØ CLOSE 14 91Ø ISAM%(1)=4 92Ø ISAM$(1)="CL" 93Ø GOSUB 6ØØØØ 94Ø IF ISAM%(8)<>Ø THEN GOTO 99Ø	L'istruzione 9ØØ chiude il file dati. Le istruzioni da 91Ø a 94Ø chiudono il file indice.

LETTURA SEQUENZIALE (RN, SN)

La funzione di lettura sequenziale ("Read Next") legge in sequenza la chiave successiva nel file e restituisce sia il nuovo valore di chiave che il numero di record corrispondente. RN e SN sono funzionalmente identiche.

La funzione di lettura sequenziale ("Read Next") permette di leggere secondo l'ordine sequenziale delle relative chiavi.

La posizione di partenza per la lettura all'interno del file può essere stabilita da una funzione di lettura con chiave ("Read Key") oppure generica ("Read Generic") oppure, se la prima funzione ISAM eseguita su un file è la lettura sequenziale ("Read Next"), la lettura parte dal primo record del file.

Se la lettura precedente era stata fatta con l'ultima chiave nell'indice, o se non ci sono chiavi nell'indice, la funzione di lettura sequenziale ("Read Next") emetterà il codice di ritorno 51 ("Not Found").

Se nell'indice ci sono chiavi duplicate, il numero di record restituito dalla funzione "Read Next", segue sempre la sequenza nella lista dei duplicati.

ISAM restituisce il numero del record in ISAM%(9); il record può quindi essere letto con l'istruzione GET del BASIC.

La chiave del record è assegnata ad ISAM$(2).

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - Numero del DCB del file
ISAM$(1) - Codice di funzione (RN o SN)

Le variabili di ritorno sono:

ISAM%(8) - Codice di ritorno
ISAM%(9) - Numero dei record dati
ISAM$(2) - Valore della chiave del record

I valori possibili del codice di ritorno sono: ØØ, 31, 41, 51 e 71.

La sequenza funzionale per leggere un record mediante la funzione di lettura sequenziale ("Read Next") è la seguente:

PASSO	OPERAZIONE
1	Aprire il file dati con l'istruzione OPEN del BASIC
2	Strutturare il buffer del file dati mediante l'istruzione FIELD del BASIC.
3	Aprire il file indice mediante la funzione "Open File"
4	Specificare il numero del DCB del file
5	Porre il codice funzione a "RN" o "SN"
6	Richiamare la subroutine ISAM
7	Controllare il codice di ritorno
8	Leggere il contenuto del record usando l'istruzione GET del BASIC.
9	Chiudere il file dati con l'istruzione CLOSE del BASIC
1Ø	Chiudere il file indice usando la funzione di chiusura di file ("Close File").

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ OPEN "R",13,"DFILE",128 11Ø FIELD 13, 128 AS A$ 12Ø ISAM%(1)=2 13Ø ISAM$(1)="00" 14Ø ISAM$(4)="INDEXFILE" 15Ø GOSUB 6ØØØØ 16Ø IF ISAM%(8)<>Ø THEN GOTO 6ØØ : : 2ØØ ISAM%(1)=2 21Ø ISAM$(1)="RN" 22Ø GOSUB 6ØØØØ 23Ø IF ISAM%(8)<>Ø THEN GOTO 7ØØ 24Ø GET 13, ISAM%(9) : : : : : : : 3ØØ CLOSE 13 31Ø ISAM%(1)=2 32Ø ISAM$(1)="CL" 33Ø GOSUB 6ØØØØ 34Ø IF ISAM%(8)<>Ø THEN GOTO 8ØØ : :	Le istruzioni 1ØØ e 11Ø aprono il file dati DFILE e strutturano il buffer. Le istruzioni da 12Ø a 16Ø aprono il file indice INDEXFILE e assegnano il valore 2 al DCB. Le istruzioni da 2ØØ a 24Ø accedono al record la cui chiave è in sequenza a quella precedentemente letta. L'istruzione 24Ø legge il contenuto del record trovato nel buffer del file. La chiave letta viene restituita in ISAM$(2). L'istruzione 3ØØ chiude il file dati. La sequenza da 31Ø a 34Ø chiude il file indice.

LETTURA SEQUENZIALE ALL'INDIETRO (RP, SP)

La funzione di lettura sequenziale all'indietro ("Read Previous") è analoga a quella sequenziale in avanti con la differenza che restituisce il record precedente anziché il successivo.

RP e SP sono funzionalmente identiche.

Il codice di ritorno 51 indica il tentativo di lettura oltre l'inizio del file.

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - Numero del DCB del file
ISAM$(1) - Codice di funzione (RP o SP)

le seguenti variabili sono restituite:

ISAM%(8) - Codice di ritorno
ISAM%(9) - Numero del record dati
ISAM$(2) - Valore della chiave.

I valori possibili del codice di ritorno sono: ØØ, 31, 41, 51 e 71.

Per leggere il record da un file dati, usando la funzione "Read Previous", occorre seguire la sequenza indicata di seguito:

PASSO	OPERAZIONE
1	Aprire il file dati mediante l'istruzione OPEN del BASIC
2	Strutturare il buffer mediante l'istruzione FIELD del BASIC
3	Aprire il file indice con la funzione di apertura di file ("Open File")
4	Specificare il numero del DCB
5	Porre il codice funzione a "RP" o "SP"
6	Richiamare la subroutine ISAM
7	Controllare il codice di ritorno
8	Leggere il contenuto del record con l'istruzione GET del BASIC
9	Chiudere il file dati usando l'istruzione CLOSE del BASIC
10	Chiudere il file indice con la funzione di chiusura di file ("Close File")

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ OPEN "R",12,"EMPFILE",128 11Ø FIELD 12,128 AS A$ 12Ø ISAM%(1)=1 13Ø ISAM$(1)="OO" 14Ø ISAM$(4)="EMPIND" 15Ø GOSUB 6ØØØØ 16Ø IF ISAM%(8)<>Ø THEN GOTO 8ØØ	Le istruzioni 1ØØ e 11Ø aprono il file dati EMPFILE e strutturano il buffer. Le istruzioni da 12Ø a 16Ø aprono il file indice EMPIND e assegnano a DCB il valore 1.
: 2ØØ ISAM%(1)=1 21Ø ISAM$(1)="SP" 22Ø GOSUB 6ØØØØ 23Ø IF ISAM%(8)<>Ø THEN GOTO 9ØØ 24Ø GET 12, ISAM%(9)	Le istruzioni da 2ØØ a 24Ø effettuano una lettura sequenziale all'indietro cioè il record dati letto dall'istruzione 24Ø ha come chiave il valore immediatamente inferiore all'ultimo letto.
: 4ØØ CLOSE 12 41Ø ISAM%(1)=1 42Ø ISAM$(1)="CL" 43Ø GOSUB 6ØØØØ 44Ø IF ISAM%(8)<>Ø THEN GOTO 95Ø	L'istruzione 4ØØ chiude il file dati. La sequenza da 41Ø a 44Ø chiude il file indice.

INSERIMENTO DI DATI IN UN FILE ISAM

INSERIMENTO ADD (WA, SA)

La funzione di inserimento in modo Add ("Write Add") inserisce una nuova chiave nel file indice, associa a tale chiave il primo numero di record disponibile e ritorna questo numero in ISAM%(9).

Per eseguire tale funzione con un file indice primario (WA), occorre aggiungere il record dati al relativo file mediante l'istruzione PUT del BASIC con riferimento al numero di record contenuto in ISAM%(9).

Se ciò non viene effettuato, la struttura del file indice non è più congruente con quella del file dati che necessita quindi di essere ricostruito.

Questa funzione può essere usata per costruire nuove strutture ISAM oppure per aggiungere record a strutture già esistenti.

Le chiavi possono essere aggiunte in qualsiasi ordine; esse vengono inserite automaticamente in sequenza corretta.

La funzione SA deve essere usata quando si esegue una funzione di inserimento di chiave in modo Add su un indice secondario.

La funzione assegna una chiave secondaria a record dati già esistenti; occorre quindi fornire ad ISAM sia il nuovo valore della chiave che il numero del record relativo.

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - Numero del DCB del file
ISAM%(5) - Numero del record (solo per SA)

ISAM$(1) - Codice di funzione (WA o SA)
ISAM$(2) - Valore della chiave

Le seguenti variabili sono ritornate al programma:

ISAM%(8) - Codice di ritorno
ISAM%(9) - Numero del record dati

I valori possibili del codice di ritorno sono: ØØ, 31, 41, 61 e 71

Note

Il programma utente che inserisce record mediante la suddetta funzione ("Write Add") deve chiudere i file usati prima di terminare la propria esecuzione.

Se ciò non viene effettuato, la struttura dei file può essere danneggiata dato che il BASIC non aggiorna i file indice fintantoché essi non vengono chiusi.

Per inserire una nuova chiave nel file indice primario usando la funzione "Write Add" e per registrare il relativo record nel file dati occorre seguire questa sequenza:

USO DI ISAM IN UN PROGRAMMA BASIC

PASSO	OPERAZIONE
1	Aprire il file dati mediante l'istruzione OPEN del BASIC
2	Strutturare il buffer mediante l'istruzione FIELD del BASIC
3	Aprire il file indice mediante l'appropriata funzione ("File Open", "Create File" o "Open/Create File")
4	Specificare il numero del DCB del file.
5	Porre il codice di funzione a "WA"
6	Specificare il nuovo valore della chiave
7	Richiamare la subroutine ISAM
8	Controllare il codice di ritorno
9	Inserire il record dati mediante l'istruzione PUT del BASIC
10	Chiudere il file dati con l'istruzione CLOSE del BASIC
11	Chiudere il file indice mediante la funzione "Close File".

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ OPEN "R",15,"DFILE-B",128 11Ø FIELD 15,128 AS A$ 12Ø ISAM%(1)=1 13Ø ISAM$(1)="00" 14Ø ISAM$(4)="INDEX-B" 15Ø GOSUB 6ØØØØ 16Ø IF ISAM%(8)<>Ø THEN GOTO 9ØØ :	Le istruzioni 1ØØ e 11Ø aprono il file dati DFILE-B e strutturano il buffer. Le istruzioni da 12Ø a 16Ø aprono il file indice INDEX-B ed assegnano il valore 1 al relativo DCB.
2ØØ ISAM%(1)=1 21Ø ISAM$(1)="WA" 22Ø ISAM$(2)="JONES" 23Ø GOSUB 6ØØØØ 24Ø IF ISAM%(8)<>Ø THEN GOTO 95Ø 25Ø PUT 15,ISAM%(9)	Le istruzioni da 2ØØ a 25Ø inseriscono nel file dati il record contenuto nel buffer e lo associano al record la cui chiave è "JONES".

PASSO	OPERAZIONE	
3ØØ CLOSE 15 31Ø ISAM%(1)=1 32Ø ISAM$(1)="CL" 33Ø GOSUB 6ØØØØ 34Ø IF ISAM%(8)<>Ø THEN GOTO 1ØØØ		L'istruzione 3ØØ chiude il file dati. La sequenza da 31Ø a 34Ø chiude il file indice.

Per inserire una nuova chiave nel file indice secondario occorre seguire la seguente sequenza:

PASSO	OPERAZIONE
1	Aprire il file dati mediante l'istruzione OPEN del BASIC
2	Strutturare il buffer mediante l'istruzione FIELD del BASIC
3	Aprire il file indice mediante l'appropriata funzione ("File Open", "Create File", "Open/Create File")
4	Specificare il numero del DCB del file
5	Specificare il numero del record
6	Porre il codice funzione a "SA"
7	Specificare la nuova chiave
8	Richiamare la subroutine ISAM
9	Controllare il codice di ritorno
10	Chiudere il file dati mediante l'istruzione CLOSE del BASIC
11	Chiudere il file indice mediante la funzione "Close File"

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ OPEN "R",15,"DFILE-B",128 11Ø FIELD 15, 128 AS A$ 12Ø ISAM%(1)=2 13Ø ISAM$(1)="00" 14Ø ISAM$(4)="INDEX-BS" 15Ø GOSUB 6ØØØØ 16Ø IF ISAM%(8)<>Ø THEN GOTO 7ØØ :	Le istruzioni 1ØØ e 11Ø aprono il file dati DFILE-B e strutturano il buffer. Le istruzioni da 12Ø a 16Ø aprono l'indice secondario INDEX-BS ed assegnano il valore 2 al DCB.
2ØØ ISAM%(1)=2 21Ø ISAM%(5)=RECNO% 22Ø ISAM$(1)="SA" 23Ø ISAM$(2)=KEY$ 24Ø GOSUB 6ØØØØ 25Ø IF ISAM%(8)<>Ø THEN GOTO 75Ø :	Le istruzioni da 22Ø a 25Ø assegnano una chiave secondaria ad un record; sia la chiave che il numero di record sono contenuti in variabili.
3ØØ CLOSE 15 31Ø ISAM%(1)=2 32Ø ISAM$(1)="CL" 33Ø GOSUB 6ØØØØ 34Ø IF ISAM%(8)<>Ø THEN GOTO 8ØØ	L'istruzione 3ØØ chiude il file dati. Le istruzioni da 31Ø a 34Ø chiudono il file indice.

FUNZIONI DI CANCELLAZIONE

CANCELLAZIONE DI CHIAVE (KD, SD)

La funzione di cancellazione di chiave ("Key Delete") permette la rimozione di una chiave dal file indice senza però rimuovere il record associato dal file dati.

KD e SD sono funzionalmente identiche.

Queste funzioni sono particolarmente utili quando un record dati dovrà essere letto in futuro ma non è più richiesto di accedere ad esso per mezzo della chiave.

Esse rimuovono fisicamente una chiave dal file indice liberando spazio per successive aggiunte. Eseguendo la funzione "Key Delete" su un file

indice a chiavi duplicate, si deve specificare il numero del record per essere sicuri che la chiave scelta sia stata cancellata. Se il numero di record vale zero, allora verrà cancellata la prima chiave trovata.

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - Numero del DCB del file
ISAM%(5) - Numero del record dati (solo per chiavi duplicate)
ISAM$(1) - Codice di funzione (KD o SD)
ISAM$(2) - Valore della chiave da rimuovere

Le seguenti variabili sono restituite:

ISAM%(8) - Codice di ritorno
ISAM%(9) - Numero del record dati

I possibili valori di ritorno sono: ØØ, 31, 41, 51 e 71.

Per cancellare una chiave dal file indice occorre seguire la seguente sequenza:

PASSO	OPERAZIONE
1	Aprire il file dati mediante l'istruzione OPEN del BASIC
2	Strutturare il buffer mediante l'istruzione FIELD del BASIC
3	Aprire il file indice usando la funzione "File Open"
4	Specificare il numero del DCB del file
5	Porre il codice di funzione a "KD" o "SD"
6	Specificare la chiave da rimuovere
7	Specificare il numero del record dati (solo per chiavi duplicate)
8	Richiamare la subroutine ISAM
9	Controllare il codice di ritorno
10	Chiudere il file dati mediante l'istruzione CLOSE del BASIC
11	Chiudere il file indice tramite la funzione "Close File"

Esempio

PROGRAMMA BASIC	COMMENTI
3ØØ OPEN "R",15,"DATA-1",128 31Ø FIELD 15,128 AS A$ 32Ø ISAM%(1)=1 33Ø ISAM$(1)="00" 34Ø ISAM$(4)="INDEX-1" 35Ø GOSUB 6ØØØØ 36Ø IF ISAM%(8)<>Ø THEN GOTO 9ØØ . .	Le istruzioni 3ØØ e 31Ø aprono il file dati DATA-1 e ne strutturano il buffer. La sequenza da 32Ø a 36Ø apre il file indice INDEX-1 e gli assegnano DCB=1.
4ØØ ISAM%(1)=1 41Ø ISAM$(1)="KD" 42Ø ISAM$(2)="1234" 425 ISAM%(5)=DATAREC% 43Ø GOSUB 6ØØØØ 44Ø IF ISAM%(8)<>Ø THEN GOTO 95Ø . .	La sequenza da 4ØØ a 44Ø cancella dal file indice la chiave "1234". L'istruzione 425 è necessaria solo se INDEX-1 contiene chiavi duplicate.
5ØØ CLOSE 15 51Ø ISAM%(1)=1 52Ø ISAM$(1)="CL" 53Ø GOSUB 6ØØØØ 54Ø IF ISAM%(8)<>Ø THEN GOTO 1ØØØ . .	L'istruzione 5ØØ chiude il file dati. La sequenza da 51Ø a 54Ø chiude il file indice.

CANCELLAZIONE DI RECORD (DR)

La funzione di cancellazione di record rimuove un record dal file dati.

La rimozione del record comporta la rimozione della chiave associata nel file indice ed il salvataggio, in cima allo stack dei record cancellati, del numero del record; questo numero verrà usato per successive funzioni di inserimento ADD ("Write Add").

Questa funzione ("Delete Record") deve essere usata, solamente con chiavi primarie; eventuali chiavi secondarie devono essere rimosse mediante la funzione di cancellazione di chiave ("Key Delete" o "Secondary Key Delete").

E' considerato un buon metodo di programmazione il riconoscere un record cancellato per mezzo di un apposito codice contenuto nel record stesso.

Questo indica che i vari record sono stati cancellati senza che sia stato riutilizzato lo spazio che si è reso disponibile.

Eseguendo una funzione "Delete Record" su un file indice a chiavi duplicate si deve specificare il numero del record, per essere sicuri che la chiave del record scelto sia stata cancellata.

Se il numero del record vale zero, allora verrà cancellata la prima chiave corrispondente.

Le seguenti variabili devono essere passate ad ISAM:

ISAM%(1) - Numero del DCB del file
ISAM%(5) - Numero del record dati (solo per chiavi duplicate)
ISAM$(1) - Codice di funzione (DR)
ISAM$(2) - Valore della chiave da cancellare

Le seguenti variabili sono ritornate al programma utente:

ISAM%(8) - Codice di ritorno
ISAM%(9) - Numero del record

I valori possibili dei codici di ritorno sono: ØØ, 31, 41, 51 e 71.

Per cancellare un record mediante la sopracitata funzione occorre seguire la sequenza:

PASSO	OPERAZIONE
1	Aprire il file dati mediante un'istruzione OPEN del BASIC
2	Strutturare il buffer in campi mediante l'istruzione FIELD del BASIC.
3	Aprire il file indice usando la funzione "File Open"
4	Specificare il numero di DCB del file
5	Porre il codice di funzione a "DR"
6	Specificare il valore della chiave

PASSO	OPERAZIONE
7	Specificare il numero del record dati (solo per chiavi duplicate)
8	Richiamare la subroutine ISAM
9	Controllare il codice di ritorno
10	Chiudere il file dati mediante l'istruzione CLOSE del BASIC
11	Chiudere il file indice mediante la funzione "Close File"

Esempio

PROGRAMMA BASIC	COMMENTI
1ØØ OPEN "R",12,RECFILE",128 11Ø FIELD 12,128 AS A$ 12Ø ISAM%(1)=3 13Ø ISAM$(1)="00" 14Ø ISAM$(4)="IFILE-B" 15Ø GOSUB 6ØØØØ 16Ø IF ISAM%(8)<>Ø THEN GOTO 6ØØ . . .	Le istruzioni 1ØØ e 11Ø aprono il file dati RECFILE e ne strutturano il buffer. La sequenza da 12Ø a 16Ø apre il file IFILE-B ed assegna il valore 3 al relativo DCB.
2ØØ ISAM%(1)=3 21Ø ISAM$(1)="DR" 22Ø ISAM$(2)="SMITH" 225 ISAM%(5)=DATAREC% 23Ø GOSUB 6ØØØØ 24Ø IF ISAM%(8)<>Ø THEN GOTO 65Ø . . .	La sequenza da 2ØØ a 24Ø cancella dal file dati il record la cui chiave è "SMITH". L'istruzione 225 legge il numero del record da una lista di variabili; è necessaria solo se IFILE-B contiene chiavi duplicate.
3ØØ CLOSE 12 31Ø ISAM%(1)=3 32Ø ISAM$(1)="CL" 33Ø GOSUB 6ØØØØ 34Ø IF ISAM%(8)<>Ø THEN GOTO 8ØØ	L'istruzione 3ØØ chiude il file dati. Le istruzioni da 31Ø a 34Ø chiudono il file indice.

A. PROGRAMMA DIDATTICO ISAM

SOMMARIO

Il programma descritto in questa appendice é rivolto ai programmatori che usano ISAM per la prima volta.

INDICE

PROGRAMMA DIDATTICO ISAM A-1

PROGRAMMA DIDATTICO ISAM

Lo scopo di questo programma è di fornire informazioni sui concetti del metodo di accesso ISAM.

Prima di usare questo programma occorre caricare in memoria i file di programma ISAM contenuti nel disco di sistema, entrare in ambiente BASIC mediante il comando BA ed avere disponibile su un "drive" un disco formattato su cui andranno i file ISAM di prova.

Il programma didattico viene richiamato dal comando:

```
run "isamx.bas"
```

Il programma richiede, in sequenza, il nome del file dati e la lunghezza del record:

```
Enter data file name: 1:dfile
Enter data record length: 128
```

Viene quindi generato un menu contenente le funzioni disponibili.

Questo menu é mostrato in Figura A-1.

```
Function Code Selections

RK,SK - Read Key                    OO - Open Existing File
RG,SG - Read Generic                OC - Create New File
RN,SN - Read Next                   OF - Create and/or Open File
RP,SP - Read Previous               CL - Close File
WA,SA - Write Add
DR    - Delete Record               X - Examine Variable
KD,SD - Key Delete
MI    - ISAM Initialize
MS    - ISAM Status
```

Figura A-1 Menu della selezione del codice di funzione

Le funzioni contenute nel menu di Figura A-1 possono ora essere provate; occorre però ricordare che la prima funzione, in ogni sessione di lavoro, deve essere sempre quella di inizializzazione (MI).

Come primo esercizio si consiglia di utilizzare l'esempio seguente.

Si noti che il numero di file 1 è riservato al file dati. Il primo numero disponibile per il DCB è 2.

VIDEO	COMMENTI
Enter desired function code or 'END': mi MI - ISAM Initialize Enter maximum number of files open: 2 Enter number of additional buffers: 2 ISAM Initialized	Durante la seguente sessione potranno essere aperti al massimo due file indice contemporaneamente. Questa (MI) deve essere la prima funzione richiamata, altrimenti viene emesso il codice di errore 31 (sequenza funzionale non corretta).
Enter desired function code or 'END':oc OC - Create Enter Index File Name: 1:IFILE1 Enter DCB number: 2 Enter Duplicate/Unique type (D/U):u Create Function successful	Viene creato ed aperto il file indice IFILE1 sull'unità 1, gli viene assegnato DCB=2 e non sono ammesse chiavi duplicate.
Enter desired function code or 'END':wa WA - Write Add Enter DCB number: 3 Enter Key value:abc Enter data record text: 123 >>Error ISAM%(8)=71 Open/Close Error	Si tenta di inserire il valore chiave 'abc' nel file indice avente DCB=3 e di scrivere il testo '123' nel relativo record dati. Questo tentativo non ha successo dato che non esiste un DCB avente valore 3 assegnato ad un file indice.
Enter desired function code or 'END': wa WA - Write Add Enter DCB number:2 Enter key value:abc Enter data record text: 123 Data Record Number = 1	Viene inserita con successo la chiave 'abc' sul file indice IFILE1, viene creato un puntatore al record dati e viene inserito il testo '123' in quel record.

VIDEO	COMMENTI
Enter desired function code or 'END': wa WA - Write Add Enter DCB number:2 Enter key value: def Enter data record text: 567 Data Record Number = 2	Viene inserita la chiave 'def' nel file indice e viene scritto '567' nel corrispondente record del file dati.
Enter desired function code or 'END': wa WA - Write Add Enter DCB number: 2 Enter key value: ØØ1 Enter data record text: ghi Data Record number = 3	In questa sezione viene eseguita un'altra funzione di scrittura Add.
Enter desired function code or 'END': rk RK - Read Key Enter DCB number: 2 Enter key value: def Data Record Number = 2 Key = def 567	Viene effettuata una lettura con chiave sul valore 'def' ottenendo il dato '567'.
Enter desired function code or 'END': rg RG - Read Generic Enter DCB number: 2 Enter key value: 1 Data Record Number = 1 Key = abc 123	Viene effettuata una lettura generica sulla chiave di valore '1'. La prima chiave maggiore o uguale a questa é 'abc' quindi si ottiene il corrispondente record di dati il cui valore é '123'.

VIDEO	COMMENTI
Enter desired function code or 'END': rn RN - Read Next Enter DCB number: 2 Data Record Number = 2 Key = def 567	Viene effettuata una lettura in avanti del valore della chiave, cioé della chiave successiva e maggiore di 'abc', che in questo caso é 'def'.
Enter desired function code or 'END': rp RP - Read Previous Enter DCB number: 2 Data Record Number = 1 Key = abc 123	Viene effettuata una lettura all'indietro del valore della chiave cioé viene letto il valore di chiave 'abc'.
Enter desired function code or 'END': oc OC - Create Enter Index File Name: 1:IFILE2 Enter DCB number: 3 Enter Duplicate/Unique type (D/U): d Create Function successful	Viene creato ed aperto un file indice secondario IFILE2 con DCB=3. Inoltre le chiavi possono essere duplicate.
Enter desired function code or 'END': sa SA - Write Add Enter DCB number: 3 Enter key value: 125 Enter data record number: 1Ø	Viene effettuata una scrittura secondaria attraverso il file indice IFILE2.
Enter desired function code or 'END': sa SA - Write Add Enter DCB number: 3 Enter key value: 125 Enter data record number: 1	Viene effettuata un'altra scrittura secondaria su IFILE2.

VIDEO	COMMENTI
Enter desired function code or 'END': ms MS - ISAM Status Enter DCB number: 3 Total Records Active: 1Ø Unused deleted records= Ø Duplicate/Unique flag = D	Viene effettuata una lettura di stato su IFILE2 per determinare quanti record attivi ci sono nel file, quanti cancellati e se il file é a chiavi uniche o duplicate.
Enter desired function code or 'END': ms MS - ISAM Status Enter DCB number: 2 Total Record Active = 3 Unused Deleted Records = Ø Duplicate/Unique Flag = U	Questa richiesta é analoga alla precedente ma su IFILE1.
Enter desired function code or 'END': cl Enter DCB number: 2 DCB Number 2 Closed	Viene chiuso IFILE1.
Enter desired function code or 'END': cl Enter DCB number: 3 DCB Number 3 Closed	Viene chiuso IFILE2.
Enter desired function code or 'END': end ISAMX Complete Ok	

B. PROGRAMMA DI CONTROLLO DEI FILE ISAM

SOMMARIO

Questo programma di utilità permette all'utente di esaminare più da vicino i file ISAM.

INDICE

PROGRAMMA DI CONTROLLO DEI FILE ISAM B-1

PROGRAMMA DI CONTROLLO DEI FILE ISAM

Questo programma ("dump utility") permette di esaminare il contenuto dei file ISAM e quindi di assicurare la corretta struttura dei programmi utente.

Prima di usare questo programma occorre caricare in memoria i file di programma ISAM contenuti nel disco di sistema, entrare in ambiente BASIC mediante il comando BA ed avere disponibile su un drive il disco contenente i file da controllare.

Il programma viene richiamato mediante:

run "isamd.bas"

L'output del programma può essere diretto su video, su stampante o su entrambi; oltre a questa opzione occorre fornire anche i nomi del file dati e del file indice.

Il programma chiede se si vuole esaminare il file indice; la risposta affermativa dà in uscita il record di testata (record Ø) del file indice seguito dalla domanda se si vogliono esaminare altri record; in questo modo è possibile leggere il numero del record.

Una volta letto il file indice, il programma chiede se si vuole esaminare il file dati.

In tal modo per ogni record viene fornito il numero del record ed il suo contenuto.

Esempio

Il programma ("dump utility"), all'inizio della sua esecuzione, emette su video una serie di informazioni (vedi fig. B-1).

In questo caso il video è stato scelto come dispositivo di uscita.

Il file indice ed il file dati scelti sono 1:IFILEA e 1:DFILEA, rispettivamente, e la lunghezza del record di dati è di 128 byte.

```
ISAM 8ØØØ - ISAM File Dump Utility
(c) Copyright 1982, OLIVETTI - All Rights Reserved

How do you wish the output to be displayed?

(1) At the console

(2) At the printer

(3) Both

Enter desired selection:1

Enter index file name: 1:IFILEA

Enter data file name: 1:DFILEA

Enter data record length: 128

Print index file? (Y/N): y
```

Figura B-1 Esempio di output del programma di dump.

Rispondendo affermativamente all'ultima domanda della figura B-1 viene visualizzato il file indice. Il programma emette quindi il contenuto del record Ø del file indice come illustrato in figura B-2.

```
INDEX FILE 1:IFILEA

HEADER CONTROL BLOCK                                        RECORD # Ø
- - - - - - - - - - - - - - - - -    - - - - - - - - - - - - - - - - - -

DCB use count                  1    Length of file name              8
Next node pointer              7    Next record pointer             22
Delete stack pointer           Ø    Root node pointer                1
Duplicate flag                 Ø    Delete stack offset              Ø
Number of levels               2    Unused deleted records           Ø

Do you wish to look at more records? (Y/N):y

Enter record number: 1
```

Figura B-2 Output dell'Header Control Block (record Ø)

Questa visualizzazione contiene informazioni sulla struttura del file indice.

Le risposte alle ultime due domande danno come uscita il contenuto del record 1, cioè il record radice, come illustrato in Figura B-3.

```
INDEX FILE 1:IFILEA
INDEX NODE                                                  RECORD # 1
- - - - - - - - - - - - - - -        - - - - - - - - - - - - - - -
Index Level                  2       Key Count                    4
PØ Pointer                   Ø

REC #  KEY LENGTH  NODE POINTER  KEY VALUE
- - -  - - - - - -  - - - - - - -  - - - - - - - - - - - - - - - -

    4          12             3   444567834523
    7          14             4   789432348765aa
   19          12             6   rem453976567
   15          16             5   u444567834523-aa

Do you wish to look at more records? (Y/N):y

Enter record number : 4
```

Figura B-3 Esempio di output di un record di nodo indice

Questo output elenca le chiavi contenute nel record con il relativo numero del record dati, la lunghezza della chiave ed il puntatore al prossimo nodo del livello inferiore.

La figura B-4 illustra il contenuto del nodo 4 che é un nodo terminale contenente 4 chiavi.

```
INDEX FILE 1:IFILEA
LEAF NODE                                                   RECORD # 4
- - - - - - - - - - - - - - -        - - - - - - - - - - - - - - - - - -
Index Level                  1       Numbers of Keys              4
Forward Pointer              6       Reverse Pointer              3

REC #  KEY LENGTH  KEY VALUE
- - -  - - - - - -  - - - - - - - - - - - - - - - - - - - - - - - - -
    7          14  789432348765aa
    8          14  889432348765ab
    9          12  995676543765
   2Ø          2Ø  q567483456823branch5

Do you wish to look at more recors? (Y/N):n
Print data file? (y/n):y
```

Figura B-4 Esempio di output di un record di nodo terminale

La risposta affermativa alle ultime due domande della figura B-4 prevede l'uscita del contenuto del file dati, come illustrato in figura B-5.

```
Data file 1:DFILEA

Data record number 1
0000000000000000000000000000000000000000

Data record number 2
0000000000000011111111111111111111111111
.
.
.
```

Figura B-5 Esempio di output di contenuto di file dati

C. VARIABILI ISAM

Pagina mancante?

VARIABILI ISAM

VARIABILI PASSATE AD ISAM

VARIABILE	DESCRIZIONE
ISAM%(1)	Numero del DCB del file
ISAM%(2)	Non usata
ISAM%(3)	Non usata
ISAM%(4)	Non usata
ISAM%(5)	Numero del record dati
ISAM%(6)	Numero di buffer addizionali
ISAM%(7)	Numero massimo di file indice aperti
ISAM%(8)	Non usata
ISAM%(9)	Non usata
ISAM$(1)	Codice di funzione
ISAM$(2)	Valore della chiave
ISAM$(3)	Non usata
ISAM$(4)	Nome del file indice
ISAM$(5)	Non usata
ISAM$(6)	Flag duplicato/unico
ISAM$(7)	Non usata
ISAM$(8)	Non usata
ISAM$(9)	Non usata

VARIABILI RITORNATE DA ISAM

VARIABILE	DESCRIZIONE
ISAM%(1)	Non usata
ISAM%(2)	Non usata
ISAM%(3)	Non usata
ISAM%(4)	Non usata
ISAM%(5)	Non usata
ISAM%(6)	Numero di record attivi
ISAM%(7)	Numero di record non usati cancellati
ISAM%(8)	Codice di ritorno
ISAM%(9)	Numero del record dati
ISAM$(1)	Non usata
ISAM$(2)	Valore della chiave
ISAM$(3)	Non usata
ISAM$(4)	Non usata
ISAM$(5)	Non usata
ISAM$(6)	Flag duplicato/unico
ISAM$(7)	Non usata
ISAM$(8)	Non usata
ISAM$(9)	Non usata

D. CODICI DI FUNZIONE

CODICE DI FUNZIONE	DESCRIZIONE
OO	Apertura di file
OC	Creazione di file
OF	Apertura/creazione di file
CL	Chiusura di file
RK	Lettura con chiave
RG	Lettura generica
RN	Lettura in avanti
RP	Lettura all'indietro
SK	Lettura su indice secondario
SG	Lettura generica su indice secondario
SN	Lettura in avanti su indice secondario
SP	Lettura all'indietro su indice secondario
WA	Scrittura Add
SA	Scrittura Add su indice secondario
DR	Cancellazione di record
KD	Cancellazione di chiave
SD	Cancellazione di chiave su indice secondario
MI	Funzione di inizializzazione
MS	Funzione di stato

E. CODICI DI RITORNO

CODICE DI RITORNO	DESCRIZIONE
ØØ	Operazione terminata con successo
31	Sequenza funzionale errata
41	Errore di sintassi
51	Record dati non trovato
61	Chiave duplicata
71	Errore in apertura/chiusura file
81	Errore di disco

AVVISO

La Ing. C.Olivetti & C.S.p.A. si riserva il diritto di apportare modifiche al prodotto descritto in questo manuale in qualsiasi momento e senza preavviso.

Questo materiale è stato preparato da Olivetti esclusivamente per l'uso da parte dei propri clienti.

Olivetti garantisce che il presente materiale costituisce, alla data di edizione, la più aggiornata documentazione da essa elaborata relativa al prodotto cui si riferisce.

E' inteso che l'uso di detto materiale avviene da parte dell'utente sotto la propria responsabilità.

Nessuna ulteriore garanzia viene pertanto prestata da Olivetti (in particolare per eventuali imperfezioni, incompletezze e/o difficoltà operative), restando espressamente esclusa ogni sua responsabilità per danni diretti o indiretti comunque derivanti dall'uso di tale documentazione.

Tutta la documentazione è coperta da copyright.